Cogneris Administrator Manual

Edition	`2026.10.1`
Generated	2026-10-28
Source commit	`bd7d967`
Online	https://docs.cogneris.ai/manuals/admin/

Contents

1. START HERE
    i. Start here
2. TENANT ADMINISTRATION
    i. Manage members and profiles
3. PORTAL
    i. Portal journey languages
4. WORKFLOW
    i. Workflow administration overview
    ii. Design and publish a workflow
    iii. Organize human review
    iv. Validate digital signatures
    v. Connect your systems
    vi. Workflow phases
    vii. Start
    viii. Crop
    ix. Split
    x. Classification
    xi. Extraction
    xii. Zero Shot
    xiii. Facematch
    xiv. Cross-check
    xv. Manual Review
    xvi. Conditional
    xvii. AI Executor
    xviii. Send Message
    xix. Generate document
    xx. Signature
    xxi. Signature validation
    xxii. REST integration
    xxiii. Send
    xxiv. KYB
    xxv. REST integration — overview
    xxvi. REST integration — Request

xxvii. REST integration — Schemas and mapping
xxviii. REST integration — Result, Test and Publication
xxix. REST integration — messages
xxx. Workflow administration screens

5. VISION
   i. Vision administration overview
   ii. Dashboard
   iii. Templates
   iv. Template editor — Schema
   v. Redaction policy
   vi. Template agent
   vii. Playground
   viii. Tenant agent settings
   ix. Developers
   x. MCP

6. TEMPLATE
   i. Template administration overview
   ii. Templates
   iii. Field sources
   iv. Render playground
   v. Word add-in

7. DOCUMENT CENTER
   i. Document Center administration overview
   ii. Triage the Inbox
   iii. Follow a process
   iv. Organize and share documents
   v. Search documents
   vi. Document Center administration screens

Start here

This manual is for the people who **set up and run Cogneris for their company**: company administrators and technical profiles. It explains how to configure each product — the languages of a Portal journey, workflows and their phases, connections to your systems, review queues, Vision's document AI, Template's document templates and the Document Center — and what every setting does.

Day-to-day use of the products — sending an access link, reviewing a document, signing, finding a document — is in the user manual. Shared topics such as sign-in, multi-factor authentication and billing are described there once and are not repeated here.

Parts of this manual

Manual	Start at	Chapters	Version	Oldest review	Owner
Tenant administration	Manage members and profiles	1	0.1.0	2026-10-08	product-documentation
Portal	Portal journey languages	1	1.0.0	2026-10-18	product-documentation
Workflow	Workflow administration overview	30	1.0.0 – 1.1.1	2026-10-09	product-documentation
Vision	Vision administration overview	10	1.0.0	2026-10-22	product-documentation
Template	Template administration overview	5	1.0.0	2026-10-22	product-documentation

Manual	Start at	Chapters	Version	Oldest review	Owner
Document Center	Document Center administration overview	6	1.0.0 – 1.2.0	2026-10-17	product-documentation

Manual edition 2026.10.1. Generated from manual.json and each chapter's front matter.

Oldest review is the earliest `last reviewed` date among the chapters of the part. Each chapter ends with its own review date, version and owner.

If a menu item described here is missing for you, your profile does not include it. Ask an administrator of your company to review your profile; see Profiles and Your team.

Last reviewed 2026-10-16 · Version 1.0.0 · Owner: product-documentation

Manage members and profiles

> **SKELETON**
>
> This chapter is a placeholder created to test the manual's structure and generation (XTRAK-2246). It describes no product behavior yet: content is written only from observed evidence, under XTRAK-2244 (inventory) and XTRAK-2250 (writing).

For tenant administrators. What a profile allows is described in the user manual's Profiles chapter.

Before you start

Pending.

Invite a member

Pending.

Change a member's profile

Pending.

Remove a member

Pending.

Audit trail

Pending.

Last reviewed 2026-10-08 · Version 0.1.0 · Owner: product-documentation

Portal journey languages

Use this procedure to choose the languages the people you invite through Portal access links may use, and to adapt the wording of their journey. Content was checked on 2026-10-09 against `portal-fe` `2563ceb` and `portal-be` `9dae623`, and observed in DEV on 2026-10-18 (`portal-fe` `00ea9a6`, `portal-be` `9ad5ae3`) with a synthetic company: the administrator published version 1 with custom wording that the recipient's page then showed, an HTML tag was refused with *HTML tags are not allowed.*, and a profile without administrator access saw the page read-only while the service refused its change. On 2026-10-22 the fields, actions and messages below were checked against the Portal interface source (`portal-fe` `01e16b4`); those details are not observed.

Who may do it

Only a **company administrator** can publish. Everyone else with access to Portal can open the page and review the settings; they see *Only tenant administrators can change journey languages and texts. You can review the settings here.*, and the service refuses their changes.

Before you start

- Until the first publication the page shows *Never published: the standard settings apply.*
- Each publication creates a new **version**. Links already sent keep working; the version a recipient reads depends on the setting **Pin complete bundles for new links** (step 4).

Steps

1. In Portal, open **Send & Collect › Journey languages**. The page reads *Loading journey languages…*, then *Choose which languages your invited visitors can use and adapt the wording of the journey for each of them.*
2. Under **Languages offered to visitors**, tick the languages to offer: **English**, **Português (Brasil)**, **Português (Portugal)**. At least one is required.
3. In **Default language**, choose one of the enabled languages. It is *used when neither the link nor the visitor's saved preference decides.* With more than one language enabled, people who create links can also fix the language of each link, and recipients see a **Journey language** selector.
4. **Pin complete bundles for new links** — when ticked, each link created after this publication keeps a fixed copy of the journey: later changes to the form or the wording

do not reach it. *Applies only to links created after this setting is published. Existing links stay unresolved.*

5. Under **Custom wording**, choose the **Language being edited**. For each message, the **Standard text** is shown; type **Your text** to replace it, or leave it empty to keep the standard. Keep every placeholder in braces exactly as shown, and stay within the length shown for the message. Security and consent notices cannot be changed: *Security and consent notice — only the standard text is shown to visitors.*
6. Check the result under **Preview**, choosing the **Preview language**.
7. Select **Publish**. The button shows **Publishing…**; *Journey languages published as version …* confirms it.

Follow the results

- **Outcomes by language** counts each link once, by the language the recipient actually saw: **Started**, **Completed**, **Abandoned**, **In progress**, **Revoked**.
- **Journeys with errors** counts links with errors reported from the recipient's page, per language.

If something goes wrong

Nothing was published. Fix the highlighted fields and try again. — nothing changed. The highlighted field says why. When the fields to fix are in a language other than the one on screen, *Other languages also have fields to fix:* names them, with a button to switch to each.

Nothing was published. Check your connection and try again. — nothing changed, and no field was refused: the page could not reach the service. Select **Publish** again.

Message	What to do
Enable at least one language.	Tick a language.
The default language must be one of the enabled languages.	Change the default, or enable that language.
There is custom wording for a language that is not enabled.	Enable the language, or clear its custom wording.
The placeholders must be the same as in the standard text. / A	Copy the placeholders from the standard text exactly.

Message	What to do
placeholder brace is missing or misplaced.	
The text is longer than allowed.	Shorten it.
Line breaks are not allowed. / *HTML tags are not allowed.*	Use plain text on one line.
Security and consent notices cannot be customized.	Clear the custom text of that message.
Only tenant administrators can publish journey languages and texts.	Ask a company administrator to publish.
This language is not supported. / *This language is listed twice.*	Reload the page and choose the languages again.
This message is not part of the journey.	The message was removed from the journey; reload the page.
The custom text is empty.	A custom text saved as empty before emptied fields were treated as "standard text". Clear the field again and publish: an empty field is no longer sent.
The journey languages could not be loaded.	Select **Retry**.

A message about custom wording appears under the text box it concerns, which gets a red border; other messages are listed above the form. Messages about the wording of a language other than the **Language being edited** are not shown: switch the **Language being edited** to find them.

Field reference

Field	What it does	Required	Format / values	Default
Languages offered to visitors	The languages recipients may use	At least one	**English**, **Português (Brasil)**, **Português (Portugal)**	As last published
Default language	*Used when neither the link nor the visitor's saved preference decides.*	Yes	One of the enabled languages	As last published
Pin complete bundles for new links	Freezes the journey of each link created afterwards. *Applies only to links created after this setting is published. Existing links stay unresolved.*	No	Box	Not ticked
Language being edited	Which language's wording the text boxes show	—	One of the enabled languages; works also in read-only mode	The first enabled language

Field	What it does	Required	Format / values	Default
Your text (one per message)	Replaces the standard text of that message. *Leave a field empty to keep the standard text. Placeholders in braces must stay exactly as shown.* Emptying a field removes the custom text, so the standard text comes back with the next publication.	No	Plain text on one line, up to the length shown (*Up to … characters*), with the placeholders listed (*Placeholders: …* or *No placeholders*)	Empty
Preview language	The language the **Preview** shows. *What the visitor reads in the selected language, with your wording applied.* The form name appears as `KYC` and dates as today.	—	One of the enabled languages; works also in read-only mode	The first enabled language

Unticking a language moves the **Default language** and the **Language being edited** to the first language still enabled. Its custom wording is kept and sent with the next publication.

Button	What happens	Who can use it	Reversible?
Publish	Publishes everything at once as a new version; reads **Publishing…** meanwhile. Nothing is published if any field is refused.	A company administrator	Publish again with the previous settings; there is no undo
Retry	Loads the page again after *The journey languages could not be*	Everyone	—

Button	What happens	Who can use it	Reversible?
	loaded.		

Area	Content
Header	*Version …*, or *Never published: the standard settings apply.*
Outcomes by language	*Counts each link once, by the language the visitor actually saw. Website visits are not included.* Columns **Language**, **Started**, **Completed**, **Abandoned**, **In progress**, **Revoked**. *No visitor has opened a link since language recording began.* when empty; *Visitor metrics could not be loaded.* on failure.
Journeys with errors	*Counts distinct links with reported errors. Translation version not recorded.* Columns **Language**, **Affected journeys**. *No visitor errors have been reported.* when empty; *Error metrics could not be loaded.* on failure.

Last reviewed 2026-10-18 · Version 1.0.0 · Owner: product-documentation

Workflow administration overview

For your company's **administrator** and **technical** profiles. What reviewers and other users do in Workflow — reviewing a document, returning it for correction, tracking a process — is described in the user manual's *Workflow* part; this part covers only what you configure and supervise.

AUDIENCE, CHANNEL AND CONTENT STATUS

- **Audience:** your company's administrator and technical profiles. Every procedure states which of the two may do it.
- **Channel:** the *Administrator Manual*, published on the documentation site at `/manuals/admin/` and also available as a consolidated HTML and PDF.
- **Source:** written from the Workflow interface and service source on 2026-10-09 (interface `08da80c`, service `28337bb`), and checked again for the pilot review on 2026-10-09 against interface `65c117e` and service `4a548b7` (REST integration phase, review queues, run notifications). It has **not yet been observed in a signed-in session**; what remains to be observed is listed in each chapter under *Not observed yet*.

What you manage in Workflow

You want to	Where	Chapter
Create a workflow, lay out its phases and put a version live	**Administration › Flow Configuration**	Design and publish a workflow
Decide who reviews which documents, and what a rejection does	**Administration › Review Queues**, the *Manual Review* phase	Organize human review

You want to	Where	Chapter
Accept a document only when the right people signed it	**Administration › Signer profiles**, the *Signature validation* phase	Validate digital signatures — *not available yet*
Deliver results to your systems and check every delivery	The *Send* phase, **Administration › Integrations & Deliveries**, **Administration › Connections**	Connect your systems
Follow volumes, processing time and reviewer workload	**Management**	Workflow administration screens

Who may do it

Workflow decides what you may do from the profiles assigned to you in your company; profiles are assigned in *Manage members and profiles*. Two profiles administer Workflow:

Profile	In Workflow
Administrator	Everything in this part. Only the administrator has **Review Queues**, **Build AI** and the **Management** reports, creates and changes **Connections** and **Signer profiles**, and sees the signing secret of a webhook.
Technical	Creates, publishes and deletes workflows, starts and follows processes, and follows and resends webhook deliveries. Can read **Connections** and **Signer profiles** but not change them.

Other profiles do not administer Workflow: an **operator** reviews documents in **Human Review**. If a menu item described here is missing for you, your profile does not include it; ask your company administrator.

> ⚠ **THE SERVICE CHECKS THE PROFILE, NOT ONLY THE MENU**
>
> Creating, publishing and deleting a workflow, reading or resending webhook deliveries, configuring review queues, the **Management** reports and changing connections are

> refused by the Workflow service to any profile not listed for them above — also when a screen is reached directly by its address.

Journeys in this part

1. Design and publish a workflow — create a workflow, add and connect phases, save a draft, publish a version, restore an older one.
2. Organize human review — add the *Manual Review* phase, assign reviewers to queues, and understand how a process goes back to its requester.
3. Validate digital signatures — register who may sign in a signer profile and check signatures with the *Signature validation* phase (*not available yet*).
4. Connect your systems — send results to a webhook, verify its signature, follow and resend deliveries, register REST connections.

Screens in this part

The Workflow administration screens chapter describes **Workflows** and the workflow editor, the publication refusals, **Review queues**, **Webhook Deliveries**, **Connections**, **Signer profiles**, the **Management** reports and **Build AI**.

Last reviewed 2026-10-09 · Version 1.0.0 · Owner: product-documentation

Design and publish a workflow

Use this journey to create a workflow, lay out its phases, and put a version live so that processes can start on it.

Who may do it: your company's **administrator** and **technical** profiles. The service refuses creating, publishing and deleting a workflow for any other profile.

Before you start

- You know which documents the workflow receives and what must happen to them — for example classify, extract, send to human review, deliver.
- If the workflow will validate signatures or deliver to your systems, prepare those first: Validate digital signatures, Connect your systems.

Create the workflow

1. Go to **Administration › Flow Configuration**. The page is titled **Workflows** and lists every workflow as a card with its name, a **Draft** or **Published** badge, its description and *v{version} · {created date}*.

2. Select **+ New Workflow**.

3. In **Create Workflow**, fill in **Name** (required) and **Description**, and choose how to **Start from**:

Option	What you get
Blank canvas	An empty graph with only the *Start* phase.
Document intake with review	Classify and extract incoming documents, then send them to human review before delivery.

Option	What you get
Identity verification	Crop the identity document, match the selfie against it and cross-check the extracted fields.
Business onboarding	Validate the company, read its documents, cross-check the data and notify the requester.
Contract signature	Extract the agreed terms, generate the contract, collect the signatures and notify the parties.

4. Select **Create**. The workflow editor opens. A template opens in **Guided** mode; a blank canvas in **Canvas** mode.

The new workflow is a **Draft**, version 0. A draft cannot be started.

Lay out the phases

The editor has two modes, switched with **Editing mode** in the toolbar.

Guided shows the flow as an ordered list of **Steps**. Use **Step type** and **Add step** to append a phase, the arrows to move a step earlier or later, and remove to delete it. *Start* cannot be moved or removed. Guided mode only works while the flow is a single chain: as soon as a phase has two outputs in use, the list is replaced by *This flow branches, so it cannot be shown as a list* and **Open the canvas**.

Canvas shows the graph.

1. Add a phase from the **Nodes** palette: click it, or drag it onto the canvas. *Start* is added for you and can exist only once.
2. Connect phases by dragging from an output handle of one phase to the next phase. A phase cannot link to itself, and the same link cannot be drawn twice.
3. Select a phase to open **Phase Configuration**: give it a **Description** (the name shown on the node) and fill in its settings.
4. To remove a phase or a link, select it and press **Delete** or **Backspace**. **Undo** and **Redo** are in the toolbar (also Ctrl/Cmd+Z, and Ctrl+Shift+Z or Ctrl+Y).

Phases you can use

Phase	What it does	Outputs
Start	Entry of every process; its output is what the process was started with.	One
Crop	Crops fields from the document. Leave **Fields to crop** empty to let the AI detect the important fields.	One
Split	Splits a file into its documents.	One
Classification	Identifies the document type. Below the **Classification Acceptance Threshold** (80 by default), or unclassified, a process takes **Alternative** — when it is connected.	**Normal** / **Alternative** — one is taken
Extraction	Extracts the fields. **Detect document type** also classifies each document and exposes the result to conditions, at the cost of one extra vision call per process.	One
Zero Shot	Extraction without a prepared template.	One
Facematch	Compares the identity documents with the selfie.	One
Cross-check	Compares a field of one document with a field of another, or with what the customer typed in the form. A divergence never rejects a case: it is sent to human review.	One
Manual Review	Stops the process until a person approves or rejects each document. See Organize human review.	One
Conditional	Asks a yes/no question about the previous phase's result and routes the process down **True** or **False**.	**True** / **False**

Phase	What it does	Outputs
Generate document	Renders the case into the document the customer signs.	One
Signature	Parks the process and asks the customer to sign what an earlier phase produced.	One
Signature validation	Checks the signatures of a PDF against a signer profile. **Not available yet** — see Validate digital signatures.	**OK** / **NOK**
REST integration	Calls your system's REST API through a connection. Its settings have the tabs **Request** (connection, **Method**, **Path**, path and query parameters, non-secret **Headers**, **Timeout (seconds)**, **Retries**, **Wait between retries (seconds)**, **Success status codes**), **Schemas**, **Request mapping**, **Response mapping** and **Test**. Transient failures (5xx, 408, 429, timeout) are retried.	**Success** / **Not found** / **Error** — with **Error** unconnected, a permanent failure or exhausted retries fail the run
Send	E-mails the previous phase's output, or posts it to your webhook when an endpoint URL is set. See Connect your systems.	One

AI Executor and **Send Message** are no longer in the palette: the Workflow service has no step that runs them, and a workflow that contains either cannot be published. See Phases that cannot run yet.

KYB is no longer in the palette: the VIES check is now a preset of the **REST integration**. A workflow that already has a **KYB** phase still opens it, with a notice that converts it.

Phases with several outputs take one of them: **Classification** (on its threshold, only when **Alternative** is connected), **Conditional**, **Signature validation** and **REST integration**. A **Classification** published with both outputs connected used to run both; it now runs one. To branch on anything else, connect the phase to a **Conditional**. See Which output is taken. Every phase has its own page, field by field, in Workflow phases.

Classification, **Extraction**, **Manual Review**, **Conditional**, **Facematch** and **Signature validation** accept only one incoming link.

Start, *Split* and *Zero Shot* have no dedicated form and show a **Config (JSON)** box instead; the service does not read it.

To see phases like these working together in a published workflow, the two videos below follow real cases end to end: an insurance claim checked by 14 cross-check rules (in Brazilian Portuguese), and a bank application caught by face match and cross-checks (in English).



Sinistro, passo a passo (3/6): As 14 regras que conferem um sinistro automaticamente (in Brazilian Portuguese) Watch on YouTube





Bank CAN on Cogneris: Catching a mule-account application with face match and cross-checks
Watch on YouTube

Save a draft

Select **Save Draft**; the toolbar confirms *Draft saved.* Saving only stores the draft: a published workflow keeps running the version that is live until you publish again.

Publish a version

1. Select **Publish**.
2. The editor checks the flow before asking you to confirm. Fix what it reports and publish again:

Message	Cause
Save the workflow first before publishing.	The workflow was never saved.
The workflow must have at least one start node. / *The workflow must have just one start node.*	There must be exactly one *Start* phase.
Signature validation "…": the OK output is not connected. (or NOK)	Both outputs of a *Signature validation* phase must lead somewhere.

3. Confirm in **Publish workflow** — *Confirm publish of version {n}?* — with **Publish**.

The editor saves the draft and publishes it. The badge changes to **Published** with the new version number.

If the service refuses, the editor shows why: in your language for **Signature validation** and **KYB** phases, otherwise the service's own message — every reason when there are several.

Failed to publish workflow. appears only when the service could not be reached or gave no reason. The refusals and their causes are listed in Workflow administration screens.

What publishing changes

- Every publish creates a **new version**, even when nothing changed, and stores an unchangeable copy of it in **Version history**.
- Processes already running keep the version they started on. Only processes started after the publish use the new version.
- If a *Send* phase has a webhook endpoint and no signing secret yet, one is created at publish.

Start a process to test it

On **Workflows**, a published workflow shows **Start**. It opens **Start Workflow**: add the **Documents to process** — PDF, images (PNG/JPG/TIFF/BMP), DOC/DOCX, XLS/XLSX, CSV or TXT, up to 50 MB each — and select **Start**. You land on **Processes**, where the new run appears.

A process started here has no requester: if a reviewer rejects its documents, it ends as failed instead of going back to anyone. See Organize human review.

Go back to an earlier version

1. In the editor, open **Version history**. Each published version shows *v{n}*, **live** on the current one, who published it and when, and its number of phases.
2. **Inspect** shows a version *as published*.
3. **Restore** copies that version into the draft. Confirm with **Copy into draft**.

Restoring does not change which version is live, and the workflow stays available: **Start** keeps running the live version, and the badge reads **Published · unpublished changes** until you **publish** the restored draft.

Delete a workflow

On **Workflows**, select the trash icon of the workflow and confirm *Delete "{name}"?* with **Delete**. The workflow leaves the list; the versions already published stay recorded for the processes that used them.

The service does not check whether processes are still running on the workflow. Delete only a workflow that has nothing in progress — see **Management › Overview**.

How to confirm it worked

- The card on **Workflows** shows **Published** and the version you confirmed.
- **Version history** lists that version as **live**.
- A process started after the publish shows the new phases on **Processes**.

If something goes wrong

What you see	What it means	What to do
Start is missing on a workflow	It is a draft — never published	Publish it.
Failed to publish workflow. or a service message	The service refused the publish	See Publication refusals.
+ New Workflow, **Publish** or **Delete** fails with an access error	Your profile is neither administrator nor technical	Ask your company administrator.

Not observed yet

- The editor has not been walked in a signed-in session; captures are pending.
- The settings of each phase are described field by field in Workflow phases, from the source only: those pages are not observed yet.

Last reviewed 2026-10-09 · Version 1.1.1 · Owner: product-documentation

Organize human review

Use this journey to put people in the loop: add a *Manual Review* phase to a workflow, decide which reviewers see which documents, and understand what happens when a reviewer rejects a document. How a reviewer decides each document is described in the user manual's *Review a document* journey.

Who may do it:

Procedure	Profile
Add the *Manual Review* phase to a workflow	**Administrator** or **technical**
Assign reviewers to queues (**Review Queues**)	**Administrator** only — the service refuses everyone else
Review documents	Any profile with **Human Review** (for example **operator**)

Add the review to a workflow

1. Open the workflow in the editor; see Design and publish a workflow.
2. Add **Manual Review** after the phases whose results a person must check (typically after *Extraction*, *Cross-check* or *Facematch*).
3. Connect its output. After a process is approved, it continues through every link of that output. To route approved processes differently, connect **Manual Review** to a **Conditional**. (A link drawn before this version on **Alternative** shows *Alternative not routed*: move it to the output before publishing again.)
4. Save and publish.

Its settings are **Skip review when every field is confident** and **Minimum confidence (%)**, described in Manual Review. You do not choose the kind of review there. The review screen adapts to the process — classification, fields, fraud, face match, KYB, split — from the documents and phases that ran before it.

When a process reaches the phase, it stops with each document *pending* in the review queue of its document type, until a reviewer approves or rejects every document.

Decide who reviews which documents

Every document waits in the **queue** of its document type. Three extra queues collect *Unidentified* documents, documents sent to *Fraud review* and documents sent to *Face conference*. When a document qualifies for *Fraud review*, that queue wins over every other one; see Documents flagged by the fraud check.

1. Go to **Administration › Review Queues**. The page lists each queue with the number of documents waiting in it.
2. Select a queue.
3. Tick the people who review it. **Find a person** filters by name or e-mail. Only members of your company are listed; there are no groups.
4. Select **Save**. *Saved* and *{n} reviewer(s) assigned* confirm it.

How assignment works:

- A reviewer **with no queue** keeps seeing every queue, including **All**.
- Once you assign someone to a queue, they see **only their queues**.
- A reviewer restricted to queues decides **document by document**: run-wide actions — reject everything, approve everything, return to the requester — are refused for them.

A queue whose document type was removed from the catalog, with nothing waiting in it, shows a warning: its assignments no longer route any document. Clear them, or rename the type back.

Documents flagged by the fraud check

The fraud check is **opt-in**: it runs during the *Extraction* phase only when your company allows it and the template used for extraction enables it. It runs in the Cogneris development environment and is being enabled in staging and production. See Fraud Detection for how to turn it on. Without it, no document goes to *Fraud review*.

The check does not answer yes or no. It gives each document a **risk level** — *Trusted*, *Needs review* or *Suspect* — built from signals such as an invalid CPF/CNPJ, a PDF saved again after creation, a broken PDF signature, or a file already submitted before.

When the *Manual Review* phase runs:

- a document at *Needs review* or *Suspect* (or one your company's enforcement policy flagged or blocked) goes to the **Fraud review** queue, whatever its document type;
- the process **cannot be approved automatically** while such a document is in it, even when every field is above the auto-approval confidence;
- a document at *Trusted* follows its normal queue.

On the review screen, a document in **Fraud review** shows a fraud panel: the risk level, the risk score, the AI-generated likelihood and **Checks performed** — each check marked *Passed*, *Flagged* or *Did not run*, with the signals behind it. The reviewer is judging whether the document is **genuine**, not whether its values were read correctly. A check that did not run examined nothing: treat it as not verified.

Assign the people who can judge authenticity to the **Fraud review** queue in **Review Queues**, the same way as any other queue.

Returning a process to its requester

What a rejection does depends on where the process came from.

Process	When every document is decided and at least one was rejected
From the Portal (submitted by a requester through an access link)	The process goes back to the requester to correct and resubmit. It appears on **Human Review › Awaiting requester correction**.
Started inside Workflow (no requester)	The process ends as failed (*manual_review_rejected*). The list of rejected documents is e-mailed to the requester's address when the run has one, otherwise to the **Recipient email** of the workflow's first *Send* phase; with neither, nobody is notified. The person who started the run is not e-mailed.

For Portal processes, reviewers also have **Return to requester** on the review screen. It sends the rejected documents back and **approves every still-pending document by omission**, without review — the confirmation says how many. It is enabled only when the process came from the Portal, at least one document was rejected and at least one is still pending.

A review that already has a rejection and gets no new decision for 24 hours (the service default) closes by itself the same way: pending documents are approved by omission and the dossier records a *System* event.

A process with an open review that nobody touches is closed automatically after an idle period (24 hours by default) **only if** it already has a rejection. A review with no rejection waits indefinitely.

How to confirm it worked

- A reviewer you assigned to a queue sees only that queue in **Human Review**.
- After a rejection on a Portal process, the process is listed on **Awaiting requester correction** with **Returned at**, **Waiting (hours)** and **Returned submission round**.
- **Management › Operator ranking** shows the documents each reviewer decided in the window.

If something goes wrong

What you see	What it means	What to do
Review Queues is missing, or it fails to load	Your profile is not administrator	Ask your company administrator.
A reviewer is told the document belongs to a queue they are not assigned to	The reviewer is restricted to other queues	Add them to the queue, or remove all their assignments so they see every queue.
A reviewer cannot reject or approve everything at once	The reviewer is restricted to queues	Decide document by document.
Return to requester is disabled	The process did not come from the Portal, nothing was rejected yet, or nothing is pending	Use **Reject** on the documents; for an internal process the run ends as failed.

What you see	What it means	What to do
Try again is missing on a process whose review was rejected	A rejection is a decision, not a failure	Start a new process with corrected documents (internal), or wait for the requester to resubmit (Portal).

Not observed yet

- Not walked in a signed-in session; captures are pending.
- The idle period is set per environment by Cogneris, not per company; 24 hours is the default value in the service.

Last reviewed 2026-10-22 · Version 1.1.0 · Owner: product-documentation

Validate digital signatures

Use this journey to accept a signed PDF only when the right people signed it with a valid ICP-Brasil certificate: register who may sign in a **signer profile**, then check documents with the **Signature validation** phase.

Who may do it:

Procedure	Profile
Create, edit and delete signer profiles	**Administrator** only. Other profiles that see **Signer profiles** can only read it.
Add the *Signature validation* phase to a workflow and publish it	**Administrator** or **technical**

NOT AVAILABLE YET

The **Signer profiles** screen and the **Signature validation** phase are in the Workflow interface, but the Workflow service that stores the profiles, tests a document and runs the phase has **not been released** (XTRAK-2326). Until it is:

- **Signer profiles** cannot load or save profiles;
- the **Test** tab cannot validate a document;
- a published workflow that reaches a *Signature validation* phase **fails** at that phase.

Do not publish a workflow with this phase until this notice is removed. This chapter describes the interface so that it is ready on release; it will be reviewed against the released service.

Before you start

- You know the CPF or CNPJ of each person or company allowed to sign. Signers are matched by the CPF or CNPJ in their ICP-Brasil certificate, never by name.
- A sample signed PDF helps: the profile can read its signers for you, and the phase can test against it.

Register who may sign

1. Go to **Administration › Signer profiles**. The list shows each profile's **Name**, **Signers** and **Usage** (in how many workflows it is used).
2. Select **New profile**.
3. Fill in **Profile name** and, optionally, **Description**.
4. Add each signer with **Add signer**:

Field	What to enter
Identifier type	**CPF (person)** or **CNPJ (company)**
CPF or CNPJ	The identifier. Check digits are verified; a sequence of repeated digits is refused.
Name (for reference only)	A name to recognise the signer. It is not used for matching.
Role	**Required** or **Optional**
Certificate type	**Any type**, **A1** or **A3**
Allowed issuing authorities	Leave empty to accept any trusted ICP-Brasil authority.

   Or select **Learn from a document (PDF)**: the ICP-Brasil signers found in the document are added as **Required**, with their issuer filled in. Review the role of each one.

5. Select **Save profile**.

Identifiers are **write-only**: once saved, a CPF or CNPJ is only ever shown masked. To change one, use **Replace** and type the new value.

Check signatures in a workflow

1. In the workflow editor, add **Signature validation** after the phase that produces the signed PDF.

2. In **Signers**, choose the **Signer profile**. **Manage profiles** opens **Signer profiles**.

3. Choose the **Rule over the signatures**:

Rule	Passes when
Any listed signer	At least one valid signature comes from a signer of the profile.
All required signers	Every signer marked **Required** signed.
N of M	At least **Minimum number of signers** different signers of the profile signed.

4. Connect **both** outputs: **OK** continues the flow; **NOK** handles a rejected document (for example a *Manual Review* or a *Send* that notifies). Publishing is refused while either output is unconnected.

5. Optional: in **Test**, choose a **PDF document** and select **Validate document**. The result is shown as a report and nothing is recorded in any run.

6. Save and publish.

The panel warns while the rule cannot work: no profile chosen, an empty profile, a minimum below one or above the number of signers, or **All required signers** on a profile with no required signer.

Read the result

The report — in the **Test** tab and in the run's phase detail — shows **Result** (OK or NOK), the reasons, and one line per signature: **Signer** (with its issuer), masked **Identifier**, **Signed at** (*Trusted timestamp* or *Declared time, not proven*) and **Status** (*Valid*, *Invalid* or *Indeterminate*).

Reason	Meaning
The document has no signature	—
The signature is not valid	The signature does not verify.

Reason	Meaning
The document was modified after signing	The content changed after the signature.
The certificate is not from a trusted authority	Not an ICP-Brasil trusted chain, or not one of the allowed authorities.
The certificate had expired at signing time	—
The certificate was revoked	—
The signer is not in the profile	A signature comes from someone not listed.
A required signer did not sign	Under **All required signers**.
Revocation could not be checked	The result is *Indeterminate*.

Delete a profile

Select **Delete** and confirm *Delete this profile?*. A profile used by a published workflow cannot be deleted: the screen lists the workflows that use it. Change or unpublish them first.

If something goes wrong

What you see	What to do
This CPF or CNPJ is not valid: the check digits do not match.	Correct the identifier.
No ICP-Brasil signature with a CPF or CNPJ was found in the document.	Add the signers by hand.
Too many tests in a short time.	Wait a moment and test again.
Publish refused: *Signature validation "…": the OK output is not connected.* (or NOK)	Connect both outputs.

What you see	What to do
Publish refused: *choose an existing signer profile*, *the signer profile has no signers*, or *the rule does not fit the signer profile*	Fix the phase's profile or rule.
New profile is missing	Your profile is not administrator.

Not observed yet

Everything in this chapter: the service is not released (see the notice above).

Last reviewed 2026-10-09 · Version 1.0.0 · Owner: product-documentation

Connect your systems

Use this journey to deliver the result of a workflow to one of your systems through a webhook, prove to that system that the call came from Cogneris, and follow every delivery. It also covers **Connections**, where you register REST endpoints and their credentials.

Who may do it:

Procedure	Profile
Configure the webhook of a *Send* phase	**Administrator** or **technical** (publishing the workflow)
See, copy and regenerate the webhook signing secret	**Administrator** only
Follow and resend deliveries (**Integrations & Deliveries**)	**Administrator** or **technical** — the service refuses everyone else
Create, edit, test and delete connections	**Administrator** only. Other profiles can only read **Connections**.

The signing secret is shared only with the people who run the receiving system — never in a ticket or an e-mail thread.

Send results to a webhook

1. In the workflow editor, add a **Send** phase where the result must leave the workflow, usually at the end.

2. In its configuration fill in:

Field	What to enter
Integration name	A name for your system, for example *My CRM*

Field	What to enter
Endpoint URL (HTTPS)	The address that receives the call. Only HTTPS on port 443 is accepted.
HTTP method	POST, PUT, PATCH or DELETE. The service always sends a POST, whatever you choose.
Description (optional)	What this integration does

With an endpoint URL set, the phase **posts the process result** to it. The panel has no field for an e-mail recipient, so set an endpoint: what the phase does without one is described in Send.

3. Optional: **Test endpoint** checks that the address answers — *Endpoint reachable.* or *Endpoint not reachable: ….*

4. Save and publish. Publishing is refused with *Send phase webhook endpoint must be a valid HTTPS URL.* when the address is not a valid HTTPS URL, and with *Send phase webhook endpoint must use the standard HTTPS port 443.* when it names another port.

Try the payload before pointing to your system

Under **Test URL**, after saving the workflow, select **Generate test URL**. Cogneris gives you a temporary address that captures everything the phase sends.

- **Use as endpoint** puts it in **Endpoint URL**; run the workflow; **View requests** lists what arrived (time, method, status, size, headers and payload).
- The URL expires after the number of days shown; its history is kept for 7 days. **Revoke** ends it earlier.
- Replace it with your real endpoint before going live.

Verify that a call came from Cogneris

Administrators see the phase's **Signing secret**, with **Copy** and **Regenerate**. It is created when an endpoint is set, and at the latest when the workflow is published.

Each call carries:

Header	Content
`X-Xtrakt-Signature`	`sha256=` followed by the hexadecimal HMAC-SHA256 of the raw request body, keyed with the signing secret
`X-Xtrakt-Run-Id`	The process (run) that produced the call
`X-Xtrakt-Workflow-Id`	The workflow
`Content-Type`	`application/json; charset=utf-8`

Your system recomputes the HMAC over the body exactly as received and compares it with the header. Use the secret **as the text shown** — do not decode it from hexadecimal.

Regenerate invalidates the previous secret at once: update the receiving system at the same time, then publish.

Follow and resend deliveries

1. Go to **Administration › Integrations & Deliveries**. The page is titled **Webhook Deliveries**.
2. Filter by **From**, **To**, **Workflow** and **Status** (*Success* or *Failed*), then **Apply**.
3. Each line shows **Date/Time**, **Workflow**, **Document**, **Endpoint**, **HTTP Status**, **Response time**, **Outcome** and **Attempt**.
4. Select a line for **Delivery detail**: the same fields plus **Run ID**, **Error message** and **Payload sent**.
5. To deliver again, select **Resend webhook**. *Webhook resent successfully.* or *Resend failed.* confirms the result. The resend uses the same HTTP method and is signed with the secret of the version that is live now — not with a secret only regenerated in the draft.

Cogneris retries a delivery automatically only when your system answers 5xx, 408 or 429. Any other 4xx answer is treated as final: fix the cause, then resend. A retry sends only the documents that did not get through; documents your system already accepted are not sent again.

Register a REST connection

Connections holds reusable base URLs and credentials for REST integrations.

> **WHO USES A CONNECTION**
>
> The **REST integration** phase of the workflow editor calls a connection; see Phases you can use. The base URL and the credentials live only in the connection, never in the phase. **Usage** lists the workflows whose phases use it.

1. Go to **Administration › Connections** and select **New connection**.
2. Fill in **Name** (up to 120 characters, unique), **Base URL** (must start with `https://`, without credentials in it, and point to a public address), **Authentication type** and **Timeout (seconds, 1 to 120)**.
3. Fill in the credentials of the authentication type:

Authentication	Fields
None	—
API key	**API key location** (Header or Query string), **Parameter name**, the key
Basic (user and password)	**Username**, password
Bearer token	The token
OAuth2 (client credentials)	**Token URL**, **Client ID**, client secret, **Scope (optional)**, **Audience (optional)**, **Client authentication**

4. Optional: **Default headers** are sent with every call. Mark a header **Secret** to store its value encrypted.
5. Select **Save**.

Secrets are **write-only**: once saved they show only as ••••• *set*. **Replace** sets a new value; leaving it untouched keeps the stored one. Secrets are stored encrypted; if they cannot be encrypted, the save is refused rather than storing them in clear.

Test connection (on a saved connection, while editing) sends one GET to the base URL with the connection's authentication and reports *Connection OK: HTTP {status} in {ms} ms* or *Connection test failed: …*. It never shows a secret or the response body. Tests are rate limited.

Delete removes the connection and wipes its secrets. A connection used by a published workflow cannot be deleted; the screen lists those workflows.

How to confirm it worked

- The test URL, or your system, received the call with a valid `X-Xtrakt-Signature`.
- **Webhook Deliveries** shows the delivery with *Success* and the HTTP status your system answered.

If something goes wrong

What you see	What it means	What to do
Enter a valid HTTPS URL.	The endpoint is not HTTPS	Use an HTTPS address.
Deliveries show *Failed* with a 4xx status	Your system refused the call; it is not retried	Fix the receiving side, then **Resend webhook**.
Your system rejects the signature	Wrong secret, or the body was changed before checking	Copy the current secret; compute the HMAC on the raw body; use the secret as text.
Signing secret is not shown	Your profile is not administrator	Ask your company administrator.
Webhook Deliveries does not load	Your profile is neither administrator nor technical	Ask your company administrator.
The URL resolves to a non-public address.	The connection points to a private network	Use a publicly reachable address.

What you see	What it means	What to do
A connection with this name already exists.	Names are unique	Choose another name.

Not observed yet

- Not walked in a signed-in session; captures are pending.
- The JSON layout of the payload is not documented here yet; use **Test URL** to see it.

Last reviewed 2026-10-09 · Version 1.1.0 · Owner: product-documentation

Workflow phases

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

A workflow is a graph of **phases**. You add them from the **Nodes** palette of the workflow editor — select one, or drag it onto the canvas — and connect each output of a phase to the phase that comes next. How to create, lay out and publish a workflow is in Design and publish a workflow; this part of the manual has one page per phase.

Content status	From source; not observed (`workflow-fe` `f88c5c1`, `workflow-be` `6703373`, 2026-10-22)

The palette

Nodes lists the phases in this order. A phase the editor cannot add at that moment is shown disabled.

Phase	Page	Outputs
Start	Start	One
Crop	Crop	One
Split	Split — no longer offered; new versions cannot be published with it	One
Classification	Classification	**Normal**, **Alternative**
Extraction	Extraction	One

Phase	Page	Outputs
Zero Shot	Zero Shot	One
Facematch	Facematch	One
Cross-check	Cross-check	One
Manual Review	Manual Review	One
Conditional	Conditional	**True**, **False**
Generate document	Generate document	One
Signature	Signature	One
Signature validation	Signature validation	**OK**, **NOK**
REST integration	REST integration	**Success**, **Not found**, **Error**
Send	Send	One

Which output is taken

Four phases choose one of their outputs once they have finished:

Phase	Outputs	Which one is taken
Classification	**Normal**, **Alternative**	**Alternative** when a document is below the **Classification Acceptance Threshold** (80 if you never changed it) or was not classified; **Normal** otherwise. Only when **Alternative** is connected: without it, the run continues through **Normal**. See Classification.

Phase	Outputs	Which one is taken
Conditional	**True**, **False**	The answer to its question.
Signature validation	**OK**, **NOK**	Its verdict.
REST integration	**Success**, **Not found**, **Error**	Its result.

For the last three, if the result cannot be read, none of their outputs is taken. Every other phase has a single output.

What changed. Before this version, **Extraction** and **Manual Review** also showed **Normal** and **Alternative**, and **Classification** never chose between its two: the phases connected to both outputs ran. Now:

- A **Classification** with both outputs connected runs **only one** of them — the one its threshold names. This applies to processes started after the change, also on versions already published.
- **Extraction** and **Manual Review** have one output. A version published before keeps running every link it has; a link that still leaves their **Alternative** output shows the badge *Alternative not routed* on the node, and a new version cannot be published until you connect that link to **Normal**.

To branch on anything else a phase found, connect it to a **Conditional**.

Phases that cannot run yet

AI Executor and **Send Message** are no longer offered in the palette: the service has no step that runs them. A workflow that contains either **cannot be published** — the editor says *"…" uses a phase that cannot run yet. Remove it before publishing.* A draft that still has one opens with the notice *This phase cannot run yet* on the node and in its settings. A version published before this change still fails at that node with *No executor registered for phase '…'*. See AI Executor and Send Message.

Pre-configured

When your company has connector presets, the palette shows them under **Pre-configured**. A preset is added as a REST integration phase with its settings filled in; its description

appears as a tooltip. If the presets cannot be loaded, the group is not shown and the rest of the palette works as usual. The VIES check, for example, is a preset — see REST integration.

KYB

KYB is no longer offered in the palette: the VIES check it performed is now a preset of the REST integration. A workflow that already has a **KYB** phase still opens it, with a notice that converts it. See KYB.

Names on the canvas

A phase added to the canvas is named after the palette label. If you keep that default name, the editor shows it in the language you are using; a name you typed stays as you wrote it.

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Start

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Start is the entry of every process. It does no work of its own: its output is what the process was started with — the documents and, for a process that comes from the portal, the data of that submission — and the phases connected to it receive that output.

Palette	**Nodes** › **Start**
Outputs	One output. *Start* has no input: nothing can link into it.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

Always. A workflow must contain exactly one *Start*, and the editor adds it for you when you create a workflow, whether from **Blank canvas** or from a template. Once a *Start* is on the canvas, **Start** is disabled in the palette, and the **Step type** list of the guided mode does not offer it.

Connect its output to the first phase that should work on the documents — for example Classification, Crop or Split. You can connect it to more than one phase; each of them receives the same documents.

What happens in a run

A process can only start on a **Published** workflow. When it starts, the service creates the *Start* step already completed and stores the process inputs as its output. The *Start* step calls no other service and does not read its own configuration.

The output holds a `documents` list. Each document carries an identifier, its name, its content type, its size and the location where the service stored it. What else it carries depends on how the process was started:

Started from	What the output carries
The **Start** button of a published workflow in **Workflows**	The uploaded documents only.
The portal	For each document, also the checklist item it answers and that item's title, its identifier, its ECM document and version, and whether it was approved in an earlier correction round. Next to the documents, a `process` block with the origin, the magic link, the correction round, the identifier, the requester's name and e-mail, the form answers and the language the visitor used.

Every phase after *Start* reads the documents from the output of the phase that leads into it. A phase that receives nothing to read falls back to these inputs.

Starting a process from Workflows

The card of a **Published** workflow has a **Start** button. It opens **Start Workflow**:

Element	What it does
Documents to process	Select one or more files. The hint reads *PDF, images (PNG/JPG/TIFF/BMP), DOC/DOCX, XLS/XLSX, CSV or TXT — up to 50 MB each.* The picker only offers those extensions.
Start	Uploads the files and starts the process. While it works the button reads **Starting…**. When the process has started, the list of runs opens.
Cancel	Closes the dialog without starting anything.

The service checks every file before it stores any of them: a file must not be empty, must be at most 50 MB, and must have one of the extensions above. It then admits each file — it checks the type, scans it against the security policy and converts it to a canonical form.

Admission completes for the whole request before the first file is stored, so a refused file leaves no partial process behind.

Configuration

Select the *Start* node to open **Phase Configuration**. Its header shows the phase name, *Start*, and the panel says *This phase has nothing to configure.*

Field	What it does	Required	Format / values	Default	Validation messages
Description	The name shown on the node.	No	Free text. Left empty, the node keeps its previous name.	*Start*	—

Common problems

Message	Cause	What to do
The workflow must have at least one start node.	The flow was saved without a *Start* (the editor no longer lets you delete it; a flow created through the API can still lack one).	Add **Start** from the palette and connect it.
The workflow must have just one start node.	The graph holds more than one *Start*.	Delete the extra one, so that one remains.
Select at least one document.	You selected **Start** in **Start Workflow** without choosing a file.	Choose at least one file.
… is larger than 50 MB.	The file you picked is over 50 MB; it is not added to the list.	Pick a smaller file, or split the document.

Message	Cause	What to do
File '…' is empty. · *File '…' exceeds the maximum allowed size of 50 MB.* · *File '…' has an unsupported extension. Allowed: …*	The service refused that file and no process was created.	Check the file against the rules above and start again.
Workflow '…' is not published. (service text)	The workflow is a draft.	Publish it — see Design and publish a workflow.
File '…' is empty. (service text)	A selected file has no content.	Remove it and choose the right file.
File '…' exceeds the maximum allowed size of 50 MB. (service text)	A file is larger than 50 MB.	Reduce the file or split it before you upload it.
File '…' has an unsupported extension. Allowed: … (service text)	The extension is not on the list above.	Convert the file to one of the listed formats.
The document type was rejected. (service text)	Admission did not accept the file's type.	Convert the file to one of the listed formats.
Document bytes do not match the declared representation. (service text)	The file's content does not match its extension.	Export the file again in the format its extension names.
The document exceeds admission limits. (service text)	The file is too large or too complex to admit.	Reduce the file and try again.
The document was rejected by the security policy. (service text)	The security scan refused the file.	Do not upload that file. Get a

Message	Cause	What to do
		clean copy from its source.
The document could not be safely canonicalized. (service text)	The file could not be converted to its canonical form.	Export the file again, for example as PDF, and retry.
The document admission quota is exhausted. (service text)	Your company reached its admission quota.	Try again later.
Document admission capacity is exhausted., *Document admission scanning is temporarily unavailable.*, *Document admission timed out.* (service texts)	Admission is temporarily unavailable.	Try again in a few moments.

Related

- Design and publish a workflow
- Classification, Crop, Split — common first phases
- Workflow screens

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Crop

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Crop cuts regions out of each document, such as a name or a document number. The Cogneris Vision service finds the regions: either the fields you list, or the fields it judges important.

Palette	**Nodes** › **Crop**
Outputs	One output.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

- At the start of an identity check, so that a person can see the relevant parts of the identity document. The **Identity verification** template starts with *Crop*, followed by Facematch, Cross-check and Manual Review.
- Whenever a reviewer should check specific regions of a document rather than the whole page.

What happens in a run

The phase reads the documents from the output of the phase that leads into it, and handles them **one at a time**:

1. A document that already failed in an earlier phase is not processed again; it is carried forward as skipped.

2. The document is sent to the Vision service. When **Fields to crop** lists fields, the request asks the service to crop only those fields; when it is empty, the service decides what to crop.
3. The crops are recorded on the document of the run.

The output is one row per document (`document`, `crops`, and `skipped` or `error` when they apply) plus the list of documents, unchanged — the next phases still work on the whole files.

The crops are kept for human review: in a Manual Review the reviewer can move, resize, add and remove crops before approving.

A process started with a Cogneris API key is sent through the Cogneris API instead of straight to the Vision service; the list of fields is passed on in the same way.

When documents fail

A failure on one document marks only that document failed; the others carry on, and later phases skip it.

- A temporary failure of the Vision service (server error, timeout, connection error, too many requests) is retried up to 3 times for that document. When the service asks to wait at most 10 seconds, the phase waits that long.
- If **no** document succeeded:
  - and a Manual Review is still ahead in the process, the phase completes anyway and the review receives the failed documents for a person to decide;
  - otherwise, if every failure was temporary, the whole phase runs again, up to 3 more times;
  - otherwise the documents are marked failed and the process fails. If exactly one Send is still ahead, it runs first to deliver the failure, every other pending phase is skipped, and the process then ends as failed.
- If at least one document succeeded, a document whose temporary failure outlasted its retries is marked failed; the phase is not run again for it.

Configuration

Select the node to open **Phase Configuration**. Its header shows the phase type as `CROPS`.

Field	What it does	Required	Format / values	Default	Validation messages
Description	The name shown on the node.	No	Free text. Left empty, the node keeps its previous name.	*Crop*	—
Fields to crop	The fields the Vision service should crop. The hint reads *Comma-separated list. Leave empty to let the AI auto-detect important fields.*	No	Field names separated by commas, for example the placeholder *Name, CPF, RG*. Spaces around each name are removed and empty entries are dropped. The names are passed to the AI as written; they are not checked against a template.	Empty — the AI detects the fields	—

Common problems

Message	Cause	What to do
Auth token missing for run; cannot call the document service. (service text, run error)	The process carries no credential for the Vision service.	Start the process again. If it persists, contact Cogneris support.
No documents found for crops. (service text, run error)	The phase received no documents.	Check that the phase leading into *Crop* passes documents

Message	Cause	What to do
		on, for example that it is linked from *Start*.
crops: no document succeeded — … (service text, run error)	Every document failed. The text lists the error of each one.	Read each document's error, check the files, then start the process again.
crops: no document succeeded because all failures were transient — … (service text)	Every document hit a temporary failure. The phase is run again automatically.	Wait. If the process then fails, start it again later.

Related

- Facematch, Cross-check — the rest of an identity check
- Manual Review and Organize human review
- Design and publish a workflow

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Split

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Split finds where each document begins and ends inside a file that holds several of them — for example one scanned PDF with an identity card, a proof of address and a payslip. The Cogneris Vision service returns the segments it found for each file.

Palette	Not offered any more: *Split* keeps the original files instead of separating them, so a new version cannot be published with it. Versions already published keep running it.
Outputs	One output.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

> **CANNOT BE ADDED TO A NEW VERSION**
>
> A draft that still has a *Split* node shows *Split does not separate documents yet* on the node and cannot be published: the service answers *The 'split' phase does not separate documents yet: its output keeps the original files. Remove the node and publish again.* Remove the node — the phases that follow receive the same files without it. Processes on versions already published keep running *Split* as described below.

When to use it

When the people who send you documents scan several of them into one file, and a person should see and confirm how that file divides.

Split does **not** turn the file into separate documents for the rest of the workflow. The phases after it still receive the original files; the segments are kept as the result of the split, for review.

What happens in a run

The phase reads the documents from the output of the phase that leads into it, and handles them **one at a time**:

1. A document that already failed in an earlier phase is not processed again; it is carried forward as skipped.
2. The file is sent to the Vision service, which returns its segments.
3. The segments are recorded on the document of the run.

The output is one row per document (`document` , `split` , and `skipped` or `error` when they apply) plus the list of documents — the same files that came in.

The split is kept for human review. In a Manual Review, the reviewer sees it as **SPLIT review**, with *… segment(s)*. That view only handles PDFs: for any other file it shows *SPLIT currently supports only PDFs*.

When documents fail

A failure on one document marks only that document failed; the others carry on, and later phases skip it.

- A temporary failure of the Vision service (server error, timeout, connection error, too many requests) is retried up to 3 times for that document. When the service asks to wait at most 10 seconds, the phase waits that long.
- If **no** document succeeded:
  - and a Manual Review is still ahead in the process, the phase completes anyway and the review receives the failed documents for a person to decide;
  - otherwise, if every failure was temporary, the whole phase runs again, up to 3 more times;
  - otherwise the documents are marked failed and the process fails. If exactly one Send is still ahead, it runs first to deliver the failure, every other pending phase is skipped, and the process then ends as failed.
- If at least one document succeeded, a document whose temporary failure outlasted its retries is marked failed; the phase is not run again for it.

Configuration

Select the node to open **Phase Configuration**. Its header shows the phase name, *Split*, and the panel says *This phase has nothing to configure.*

Field	What it does	Required	Format / values	Default	Validation messages
Description	The name shown on the node.	No	Free text. Left empty, the node keeps its previous name.	*Split*	—

Common problems

Message	Cause	What to do
Auth token missing for run; cannot call the document service. (service text, run error)	The process carries no credential for the Vision service.	Start the process again. If it persists, contact Cogneris support.
No documents found for split. (service text, run error)	The phase received no documents.	Check that the phase leading into *Split* passes documents on.
split: no document succeeded — … (service text, run error)	Every document failed. The text lists the error of each one.	Read each document's error, check the files, then start the process again.
split: no document succeeded because all failures were transient — … (service text)	Every document hit a temporary failure. The phase is run again automatically.	Wait. If the process then fails, start it again later.
SPLIT currently supports only PDFs.	In the review, the document is not a PDF.	Ask for the documents as PDF.

Message	Cause	What to do
Later phases still process the whole file	*Split* does not create separate documents for the next phases.	Ask senders to upload one document per file when later phases must treat them separately.

Related

- Manual Review and Organize human review
- Start
- Design and publish a workflow

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Classification

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Classification identifies the type of each document of the process. It sends the documents to the Cogneris Vision service, which returns a document type, a confidence and, when one of your company's templates covers that type, the template.

Palette	**Nodes** › **Classification**
Outputs	**Normal** and **Alternative**. **Alternative** is taken when a document is below the acceptance threshold or was not classified — only when it is connected. See Outputs.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

- Before Extraction, so that extraction follows the fields your template declares instead of letting the AI name them.
- Before a Manual Review whose queues are organised by document type: classification records each document's type and template, and the review uses it.
- Before a Conditional that routes on the document type.

If you only need the type to route right after extraction, the **Detect document type** option of Extraction does it without a separate phase.

Classification accepts only one incoming link. The editor does not draw a second one.

What happens in a run

1. The phase reads the documents from the output of the phase that leads into it (from *Start*, these are the process documents). Documents that already failed in an earlier phase are left out and carried forward as skipped.
2. It sends all remaining documents to the Vision service **in one call**.
3. For each document it records the result: the **document type**, the **confidence**, the **pages** and the **template** of your company that covers that type. The template is empty when no template covers the type. Results are matched to documents by file name, or by order when the service returns no name.
4. It stamps the type and the template on the document of the run. Manual review uses this to send each document to the queue of its template type.
5. The output is one result row per document (`document`, `classification`, and `skipped` for a document left out) plus the list of documents, unchanged. The next phase works on the same files.

A later phase or condition reads these values at `results[N].classification.documentType`, `results[N].classification.confidence` and `results[N].classification.templateId`, where `N` is the document's position starting at 0.

When every document had already failed, the phase completes without calling the Vision service and every row is marked skipped.

Failures

Classification is a single call for all documents, so an error fails the phase as a whole — never a single document.

- A temporary failure of the Vision service (server error, timeout, connection error, too many requests) is retried up to 3 times inside the phase. When the service asks to wait at most 10 seconds, the phase waits that long. If it still fails, the phase is run again, up to 2 more times.
- An error answered by the Vision service, or a permanent refusal, fails the phase and the process at once.

Outputs

The node has two outputs, **Normal** and **Alternative**. When *Classification* completes, the run takes **one** of them:

Result	Output taken
Every classified document has a confidence at or above the **Classification Acceptance Threshold**	**Normal**
At least one document is below the threshold, or the classifier returned nothing for it	**Alternative**
Alternative is not connected	**Normal**, whatever the confidence — the threshold only routes when there is somewhere to go

The phases on the output not taken are skipped. A link drawn without choosing an output counts as **Normal**. Documents that already failed before this phase do not count.

What changed. Before this version, the service did not read the threshold and ran the phases on both outputs. A published workflow whose *Classification* has both outputs connected now runs only one of them.

To route on the document type, or on anything other than confidence, connect *Classification* to a Conditional and branch on its **True** and **False** outputs.

Configuration

Select the node to open **Phase Configuration**. Its header shows the phase type as `CLASSIFICATION`.

Field	What it does	Required	Format / values	Default	Validation messages
Description	The name shown on the node.	No	Free text. Left empty, the node keeps its previous name.	*Classification*	—

Field	What it does	Required	Format / values	Default	Validation messages
Classification Acceptance Threshold	The minimum confidence for a document to go on through **Normal**. The hint reads *Minimum confidence (0–100%; 80 if left untouched). A document below it, or not classified, takes the Alternative output — only when Alternative is connected; otherwise the run continues through Normal.* It does not change the classification itself, only the output taken.	No	Whole or decimal number. A value below 0 is saved as 0, above 100 as 100. Left empty, the field is removed.	80 — also when the field was never changed and nothing is saved.	—

There is no template or document type to choose here. The Vision service decides the type and returns the template of your company that covers it.

Common problems

Message	Cause	What to do
(none — the link is not drawn)	You tried to connect a second phase into *Classification*. It accepts one incoming link.	Remove the existing link first, or place the phases in sequence.
Auth token missing for run; cannot call the document service. (service text, run error)	The process carries no credential for the Vision service.	Start the process again. If it persists, contact Cogneris support.
No documents found for classification. (service text, run error)	The phase received no documents.	Check that the phase leading into *Classification* passes documents on, for example that it is linked from *Start*.
Classifier call failed: … (service text, run error)	The Vision service answered an error, or stayed unavailable after the retries.	Check that your company has templates for the documents you receive. If the text names a temporary failure, start the process again.
A document has an empty template	No template of your company covers the type that was detected.	Create or adjust a template for that document type. Without it, Extraction names the fields itself.
Every process goes to **Alternative**	The threshold is above the confidence the classifier gives your documents (80 unless you changed it).	Lower **Classification Acceptance Threshold**, or check the classifier's training.

Message	Cause	What to do
A low-confidence document went on through **Normal**	**Alternative** is not connected, so the threshold does not route.	Connect **Alternative** to the phase that should handle those documents.

Related

- Extraction — uses the template found here
- Zero Shot — type and content without a template
- Conditional — route on the type or the confidence
- Organize human review
- Design and publish a workflow

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Extraction

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Extraction reads the fields of each document of the process. It sends every document to the Cogneris Vision service and keeps the fields it returns. When an earlier Classification found the template of a document, the extraction follows that template's fields.

Palette	**Nodes** › **Extraction**
Outputs	One. See Outputs.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

Whenever a later phase needs the content of the documents: a Manual Review where a person checks the fields, a Cross-check, a Conditional on a field value, Generate document, or a delivery with Send or REST integration.

Put a Classification before it when your documents have templates: the fields then have the names your template declares. Without it the AI names the fields itself, and the names can differ from one process to the next.

Extraction accepts only one incoming link. The editor does not draw a second one.

What happens in a run

The phase reads the documents from the output of the phase that leads into it, and handles them **one at a time**:

1. A document that already failed in an earlier phase is not processed again; it is carried forward as skipped.
2. If an earlier phase resolved the document's template, the call asks the Vision service for exactly that template's fields. Otherwise the Vision service decides which fields to return.
3. The fields are recorded on the document of the run, and the document counts as succeeded — even when the service found no value at all, as on a blank page.
4. What an earlier Classification or Zero Shot said about the document (its `classification` or `zeroShot` result) is copied into the row, so phases further on still see the type.

The output is one row per document (`document`, `fields`, the type information carried forward, and `skipped` or `error` when they apply) plus the list of documents, unchanged.

A process started with a Cogneris API key is sent through the Cogneris API instead of straight to the Vision service. On that path the template is not passed on, so the fields are always named by the AI. **Detect document type** works on that path too: the classifier is reached through the Cogneris API as well.

Detect document type

With **Detect document type** on, after the fields are read the phase sends the documents that did not fail to the Vision classifier, in one extra call. For each document it adds:

- `classification` — the document type, the confidence, the pages and the template that covers the type, as Classification produces them. It replaces any classification carried from an earlier phase.
- `identified` — `true`, or `false` when no type was found or the type is `UNKNOWN`.

It also stamps the type and template on the document of the run, as *Classification* does, so a later Manual Review can file the document under its type.

This step never fails the phase. If the classifier answers an error, the rows are left without a type; a Conditional that reads them then finds nothing.

A Conditional placed after *Extraction* can read these values at `results[N].identified` and `results[N].classification.documentType`, where `N` is the document's position starting at 0. Extracted values are under `results[N].fields`.

When documents fail

A failure on one document marks only that document failed; the others carry on, and later phases skip it.

- A temporary failure of the Vision service (server error, timeout, connection error, too many requests) is retried up to 3 times for that document. When the service asks to wait at most 10 seconds, the phase waits that long.
- If **no** document succeeded:
  - and a Manual Review is still ahead in the process, the phase completes anyway and the review receives the failed documents for a person to decide;
  - otherwise, if every failure was temporary, the whole phase runs again, up to 3 more times;
  - otherwise the documents are marked failed and the process fails. If exactly one Send is still ahead, it runs first to deliver the failure, every other pending phase is skipped, and the process then ends as failed.
- If at least one document succeeded, a document whose temporary failure outlasted its retries is marked failed; the phase is not run again for it.

Outputs

The node has one output: every phase linked to it runs once *Extraction* completes.

Before this version it also showed **Alternative**, which the service never chose between: both outputs ran. A version published with a link on **Alternative** keeps running it. In the editor, such a link keeps its dot and the node shows *Alternative not routed*; connect the link to the output before you publish a new version — the publication is refused with *"…" has an edge on its Alternative output, which only Classification routes.*

To route on whether the document was identified, or on a field value, connect *Extraction* to a Conditional.

Configuration

Select the node to open **Phase Configuration**. Its header shows the phase type as `EXTRACTION`.

Field	What it does	Required	Format / values	Default	Validation messages
Description	The name shown on the node.	No	Free text. Left empty, the node keeps its previous name.	*Extraction*	—
Detect document type	Also classifies each document and adds its type, confidence and `identified` flag to the result. The hint reads *Classifies each document during extraction and adds its type, its confidence and whether it was identified to the extraction output. Costs one extra vision call per process.*	No	On / off	Off	—

There is no template to choose here: the template comes from an earlier Classification. The type found by **Detect document type** arrives after the fields are read, so it serves routing and review but does not change the extraction it belongs to.

Common problems

Message	Cause	What to do
(none — the link is not drawn)	You tried to connect a second phase into *Extraction*. It accepts one incoming link.	Remove the existing link first, or place the phases in sequence.
Auth token missing for run; cannot call the document service. (service text, run error)	The process carries no credential for the Vision service.	Start the process again. If it persists, contact Cogneris support.
No documents found for extraction. (service text, run error)	The phase received no documents.	Check that the phase leading into *Extraction* passes documents on.
extraction: no document succeeded — … (service text, run error)	Every document failed. The text lists the error of each one.	Read each document's error. Check the files and your templates, then start the process again.
extraction: no document succeeded because all failures were transient — … (service text)	Every document hit a temporary failure. The phase is run again automatically.	Wait. If the process then fails, start it again later.
Field names change between processes	No template was found for the document, so the AI named the fields.	Add a Classification before *Extraction* and make sure a template covers the document type.
No type in the result although **Detect**	The classifier answered an error (the run log has *detectDocumentType classifier returned errors*), or the API	Check the classifier, or the API key's access.

Message	Cause	What to do
document type is on	key used to start the process cannot use the classifier.	
Publishing says the node *has an edge on its Alternative output*	A link drawn before this version still leaves **Alternative**.	Connect it to the node's output (**Normal**), or route with a Conditional.

Related

- Classification — resolves the template extraction follows
- Zero Shot — type and content without a template
- Conditional — route on the result
- Manual Review and Organize human review
- Design and publish a workflow

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Zero Shot

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Zero Shot reads documents for which you have no prepared template. For each document, the Cogneris Vision service decides on its own what type of document it is and what it contains.

Palette	**Nodes** › **Zero Shot**
Outputs	One output.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

- For documents that no template of your company covers — for example a type you receive too rarely to model.
- After a Conditional that catches documents Extraction could not identify (with its **Detect document type** option on), so that only those go through *Zero Shot*.

When the documents have templates, prefer Classification followed by Extraction: the fields then have the names your templates declare.

What happens in a run

The phase reads the documents from the output of the phase that leads into it, and handles them **one at a time**:

1. A document that already failed in an earlier phase is not processed again; it is carried forward as skipped.
2. The document is sent to the Vision service without any template.
3. The answer — the **document type**, the **confidence** and the content the model read — is recorded on the document of the run, and the type is stamped on it, so a later Manual Review can file the document under that type.

The output is one row per document (`document`, `zeroShot`, and `skipped` or `error` when they apply) plus the list of documents, unchanged. A later phase or condition reads the type at `results[N].zeroShot.documentType`, where `N` is the document's position starting at 0.

Zero Shot never resolves a template. An Extraction placed after it copies its result forward but still lets the AI name the fields.

In the phase detail of a run, *Zero Shot* shows the type found for each document. The content the model read is not shown there, because it is free-form and may hold personal data.

When documents fail

A failure on one document marks only that document failed; the others carry on, and later phases skip it.

- A temporary failure of the Vision service (server error, timeout, connection error, too many requests) is retried up to 3 times for that document. When the service asks to wait at most 10 seconds, the phase waits that long.
- If **no** document succeeded:
  - and a Manual Review is still ahead in the process, the phase completes anyway and the review receives the failed documents for a person to decide;
  - otherwise, if every failure was temporary, the whole phase runs again, up to 3 more times;
  - otherwise the documents are marked failed and the process fails. If exactly one Send is still ahead, it runs first to deliver the failure, every other pending phase is skipped, and the process then ends as failed.
- If at least one document succeeded, a document whose temporary failure outlasted its retries is marked failed; the phase is not run again for it.

Configuration

Select the node to open **Phase Configuration**. Its header shows the phase name, *Zero Shot*. The phase has nothing to configure beyond its **Description**: the panel says *This phase has nothing to configure.*

Field	What it does	Required	Format / values	Default	Validation messages
Description	The name shown on the node.	No	Free text. Left empty, the node keeps its previous name.	*Zero Shot*	—

Common problems

Message	Cause	What to do
Auth token missing for run; cannot call the document service. (service text, run error)	The process carries no credential for the Vision service.	Start the process again. If it persists, contact Cogneris support.
No documents found for zero-shot. (service text, run error)	The phase received no documents.	Check that the phase leading into *Zero Shot* passes documents on.
zero_shot: no document succeeded — … (service text, run error)	Every document failed. The text lists the error of each one.	Read each document's error, check the files, then start the process again.
zero_shot: no document succeeded because all failures were transient — … (service text)	Every document hit a temporary failure. The phase is run again automatically.	Wait. If the process then fails, start it again later.
Field names differ from one process to the next	*Zero Shot* uses no template, so the AI names the fields.	If the document type recurs, create a template for it and use Classification and Extraction.

Related

- Classification and Extraction — the templated path
- Conditional — send only unidentified documents to *Zero Shot*
- Manual Review
- Design and publish a workflow

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Facematch

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Compares the selfie of a process with the identity documents it must belong to, and publishes one verdict per document and one decision for the process. A missing selfie, an unreadable face or a failed comparison never fails the run: it sends the process to human review.

Palette	**Nodes** › **Facematch**
Outputs	One output. The phase never branches; it writes a decision (`pass` , `review` or `notApplicable`) that a later phase reads.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

- Identity checks (KYC): the customer sends an ID and a selfie, and you need to know they are the same person.
- After a **Classification** or **Zero Shot** phase — Facematch needs the document types those phases assign.
- Before **Manual Review**: the face conference is shown to the reviewer, and a manual review never approves on its own when a Facematch ran.

The **Identity verification** workflow template includes this phase. A face comparison can also be a rule of a Cross-check (**Face match**); the dedicated phase adds a threshold, explicit targets and per-document detail.

What happens in a run

Inputs. The documents of the run, and the document type each one received from an earlier **Classification** or **Zero Shot** phase.

1. **The selfie is found.** The selfie is the document whose type equals **Selfie document type** (`selfie` when the field is empty). If no document has that type — the normal case, since templates rarely type a selfie — the service looks for a face on the run's images whose type could not be identified, up to 5 images. Exactly one image with a face becomes the selfie, and its type is set to the selfie type for the rest of the run.

2. **The targets are chosen.** With **Documents that must match the selfie** ticked, only documents of those types are compared; a ticked type the process did not supply is reported as `document_not_present`. With nothing ticked, the selfie is compared with every other document of the run.

3. **Each target is compared**, one call per document (each comparison is billed). The result for that document is:

Result of the comparison	Verdict	Reason
Same person: score at or above **Match threshold**, or — with no threshold — the face service's own verdict	`match`	—
Different person	`mismatch`	`different_person`
No face found in the document	`inconclusive`	`no_face_in_document`
More than one face in the document	`inconclusive`	`too_many_faces`
The face service answered with an error, or the call failed	`inconclusive`	`facematch_error: …`

Result of the comparison	Verdict	Reason
The document failed in an earlier phase	`inconclusive` (or `notApplicable` when skipping)	`document_failed_upstream`

4. **Cases without a comparison** produce a single verdict instead:

Situation	Verdict	Reason
No selfie found	`inconclusive`, or `notApplicable` when **Skip when the selfie or the document is missing** is on	`selfie_not_present`
More than one image with a face	`inconclusive`	`multiple_selfies`
No target document	`inconclusive`, or `notApplicable` when skipping	`no_target_document` / `document_not_present`
The face-detection service did not answer for an image after the retries	`inconclusive`	`selfie_probe_unavailable`

5. **The process decision** is:

   - `pass` — at least one document matched and none is `mismatch` or `inconclusive`;
   - `notApplicable` — nothing was compared (every verdict is `notApplicable`); this is not a pass, it means the process was never face-checked;

- `review` — anything else.

Outputs. A `facematch` block in the run's data: `decision`, `selfie` (its file name), `threshold`, a `summary` (`total`, `match`, `mismatch`, `inconclusive`, `notApplicable`) and one entry per check (document, document type, verdict, score, whether a face was found, whether there were too many faces, reason). Each compared document also gets its own result (`results[…].facematch`, with `score` and `matched`), which the review screen shows next to the document.

Retries and failures. When the face-detection service refuses temporarily while looking for the selfie, the phase is run again after 30 seconds, then after 2 minutes. The run fails only for a structural problem: no documents, no document types from an earlier phase, or a selfie that cannot be read from storage.

To route on the decision, add a **Conditional** that tests `facematch.decision` — for example **=** `pass` to demand a successful check, or **≠** `review` to also let `notApplicable` through.

Configuration

Field	What it does	Required	Format / values	Default	Validatio message
Description	Name shown on the node.	No	Text	*Facematch*	—
Selfie document type	The document type that is the selfie. Every other compared document is matched against it.	No	Text; compared ignoring case	`selfie` (shown as placeholder)	—
Documents that must match the selfie	One checkbox per template of your company. Only the ticked types are compared.	No	Template names; matched against the document type an	None ticked — compare with every other document	*Could not load your templates so targets cannot be picked. With none*

Field	What it does	Required	Format / values	Default	Validatio message
			earlier phase assigned		*checked, every oth document is compared*
Match threshold (0-1)	The minimum score for a match, for this workflow only.	No	Number from 0 to 1, step 0.01; values outside are brought back into range	Empty — the face service's own verdict	—
Skip when the selfie or the document is missing	Off: a missing selfie or target sends the process to review. On: it is recorded as `notApplicable`.	No	On / off	Off	—

Common problems

Message	Cause	What to do
Could not load your templates, so targets cannot be picked. With none checked, every other document is compared.	The template list did not load, so no target can be ticked.	Reload the editor.

Message	Cause	What to do
No documents found for facematch. (run error)	The run reached the phase with no documents.	Check what the process was started with.
Facematch requires an upstream classification or zero-shot phase that assigns document types. (run error)	No **Classification** or **Zero Shot** ran before the phase.	Add one before **Facematch**.
facematch: could not read selfie '…': … (run error)	The selfie file could not be read from storage.	Start the process again; if it repeats, contact support.
Decision `review` with reason `selfie_not_present`	No document has the selfie type and no unidentified image with a face was found.	Check **Selfie document type**, or that the form asks for a selfie.
Decision `review` with reason `multiple_selfies`	More than one unidentified image has a face.	A reviewer decides.
Decision `notApplicable`	**Skip when the selfie or the document is missing** is on and nothing could be compared.	Turn it off for identity checks that must always run.

Related

- Cross-check — **Face match** rules.
- Classification, Zero Shot — assign the document types this phase needs.
- Manual Review — selfies wait in their own review queue.
- Conditional
- Organize human review

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Cross-check

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Compares values of a case with each other — a field of one document against a field of another document, against what the customer typed in the form, or against process data — and publishes a verdict for each rule and for the whole case. A divergence never rejects a case: it marks the case for human review.

Palette	**Nodes** › **Cross-check**
Outputs	One output. The phase never branches; it writes a decision (`pass` or `review`) that a later phase reads — see Use the decision.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , comparison rules from `vision-be` `fce7fcc` , 2026-10-22)

When to use it

- After **Extraction** (or **Zero Shot**), when two documents of the same case must agree — the name on the ID and on the proof of address, the amount on the proposal and on the payslip.
- When a document must agree with the form — the document number the customer typed against the one read off the ID.
- Before **Manual Review**, so the reviewer sees which rules diverged and on which documents.

The **Identity verification** and **Business onboarding** workflow templates include this phase.

What happens in a run

1. **The case is gathered.** The phase reads the documents of the run with their type, template, checklist slot and extracted fields, the form answers, and the process data. Extracted fields are addressable by their full dotted path and by their short name (for example both `caderneta.distrito` and `distrito`). Placed after a **Manual Review**, the phase still finds the documents and fields of the phases before the review.
2. **Each rule is resolved into pairs of values.** For each side (**Compare** and **With**):
   - **Document**: every document of the case that fills the chosen role. The role is matched by checklist slot first, then by template, then by classified document type. If several documents fill it, each pairing is a separate comparison (a document is never compared with itself). At most **10 pairings per rule** are compared; when a rule has more, the rest are dropped and the rule's reason says `truncated_to_10_combinations`.
   - **Form answer**: the value of that form answer.
   - **Process data**: the value of that process field. Process data includes `identifier`, `requester.name` and `today` (the day the check runs, `yyyy-MM-dd`).
3. **The pairs are compared.** Every pair is compared by its rule's **Comparison** (see Comparisons). All non-face rules go to the comparison service in a single call; face rules go to the face-matching service, one call per pairing.
4. **Each rule gets one verdict**: `match`, `mismatch`, `inconclusive` or `notApplicable`, combined from its pairings as described in How a rule's verdict is combined.
5. **The case gets one decision**: `pass` when no rule is `mismatch` or `inconclusive`; otherwise `review`. Rules that are `notApplicable` do not prevent a `pass`.
6. **Implicated documents are marked.** Every document involved in a rule that did not match is recorded with the ids of those rules, so the review screen can show the reviewer why the document is in front of them.

The phase adds a `crossCheck` block to the run's data: `decision`, a `summary` (`total`, `match`, `mismatch`, `inconclusive`, `notApplicable`), and one entry per rule in `checks` (rule id, name, comparison, verdict, score, method `deterministic` or `ai`, reason, and the values compared). When the case has face rules and no earlier **Facematch** phase ran, the face results are also published as a `facematch` block, so the review screen can show the face conference.

Comparisons

Comparison	How two values are compared	Can Let AI settle close calls decide it?
Person name	Name particles are ignored. The same names, or the same names in another order, match. When one name is contained in the other (also abbreviated, such as an initial for a first name), or they share at least half their names, the pair is a close call. Otherwise they do not match. **Accept a surname changed by marriage** is passed with the pair; the first, rule-based pass does not use it, so it only matters for close calls settled by AI.	Yes
Text	Equal when they are the same ignoring case, accents and extra spaces. Any other difference is a close call.	Yes
Exact	Character-for-character equality. No normalization.	No
Document number	Only the digits are compared. Equal digits match. A masked number matches when it shows at least 4 digits and they are the end of the other number. Other differences do not match. A value with no digits is `inconclusive`.	No
Date	Both values are read as dates; the same day matches, a different day does not. A date that cannot be read (for example written out in words) is a close call.	Yes
Amount	Both values are read as numbers, in either `1.234,56` or `1,234.56` style, ignoring currency symbols. They match when the difference is within the tolerance: the absolute tolerance when it is set (it wins), otherwise the percentage of the larger value, otherwise zero. A value that cannot be read is `inconclusive`.	No

Comparison	How two values are compared	Can Let AI settle close calls decide it?
AI judgement	Always decided by AI, using the question you write for the pair.	Always
Face match	Compares the face on the **Compare** document with the face on the **With** document. Put the selfie on **With**. Both sides must be documents. No face found, more than one face, or a failed call is `inconclusive`.	No

A side with no value — the document does not carry the field — makes the pair `inconclusive`, never a mismatch.

A close call is decided by AI when **Let AI settle close calls** is on. When it is off, close calls come back `inconclusive` and go to review.

How a rule's verdict is combined

Situation	Rule verdict
A side names a document role that no document of the case fills, and **When the document is missing** is **Send for review**	`inconclusive` (reason `role_not_present`)
The same, and **When the document is missing** is **Skip the rule**	`notApplicable`
When several documents match is **Any one is enough**	`match` if any pairing matches; otherwise `mismatch` if any pairing does not match; otherwise `inconclusive`

Situation	Rule verdict
When several documents match is **All must agree**	`mismatch` if any pairing does not match; otherwise `inconclusive` if any pairing is not a match; otherwise `match`
When several documents match is **Only the first**	Only the first document on each side is used, so the rule has one pairing and takes its verdict
The comparison service refuses or fails the call after its retry	Every rule of that call is `inconclusive`, with the service's reason

When the document is missing applies only to a missing document role. A document that is present but lacks the field is a missing value, which is `inconclusive` whatever this option says.

The rule reports the pairing that decided its verdict, and lists every pairing tried as evidence.

Use the decision

The phase has a single output: the run always continues to the next phase. To treat a `review` differently, either place a **Manual Review** after it (a manual review that would approve on its own never does so when a cross-check ran — see Manual Review), or add a **Conditional** that tests the field `crossCheck.decision`.

When the phase fails

The phase fails the run only for a structural problem: no rules, or no documents. A temporary failure of the comparison service is retried once; after that, the rules are recorded as `inconclusive` instead of failing.

Examples

Three cases, each with the rules a company would set up. The field names are illustrative: yours are the field names of your own extraction templates and the keys of your own form.

Bank: opening an account remotely

A customer opens an account from their phone. The form asks for their name, document number, date of birth and address; they send their identity card, a proof of address and a live

selfie. The bank wants to stop mule accounts — an applicant using someone else's documents, or documents that do not belong together.

Rule name	Compare	With	Comparison	Settings	
Name on the ID	**Document** Identity card › `full_name`	**Form answer** `fullName`	**Person name**	—	T t r s e N a c p c r
Document number	**Form answer** `documentNumber`	**Document** Identity card › `document_number`	**Document number**	—	A i r a c r r r v 4 c
Date of birth	**Document** Identity card › `birth_date`	**Form answer** `birthDate`	**Date**	—	A c f c l

Rule name	Compare	With	Comparison	Settings	
Address	**Document** Proof of address › `address`	**Form answer** `address`	**Text**	**Let AI settle close calls** on	A a A s f a c s
Proof of address holder	**Document** Proof of address › `holder_name`	**Document** Identity card › `full_name`	**Person name**	**Accept a surname changed by marriage** on	A s e A s e a
Selfie	**Document** Identity card	**Document** Selfie	**Face match**	—	T t r p l

When every rule matches, the decision is `pass` and the case continues without a person. When the face does not match, or any rule diverges, the decision is `review`: the case goes to the **Manual Review** that follows, and the reviewer sees which rule diverged and on which documents.





Bank CAN on Cogneris: Catching a mule-account application with face match and cross-checks
Watch on YouTube

Insurer: an insurance claim

A policyholder reports a claim through the Portal and sends the policy, the incident report, the repair invoices, their identity card and a bank statement for the payment. The insurer wants to pay clear claims quickly and send the doubtful ones to an adjuster.

Rule name	Compare	With	Comparison	Settings
Policyholder	**Document** Policy › `holder_name`	**Form answer** `fullName`	**Person name**	—
Policy number	**Form answer** `policyNumber`	**Document** Policy › `policy_number`	**Document number**	—
Incident date	**Form answer** `incidentDate`	**Document** Incident report › `incident_date`	**Date**	—
Policy in force	**Document** Policy › `coverage_period`	**Form answer** `incidentDate`	**AI judgement**	*Is the incident*

Rule name	Compare	With	Comparison	Settings
				date inside the coverage period of the policy?
Amount claimed	**Form answer** `amountClaimed`	**Document** Repair invoice › `total`	**Amount**	**Tolerance (absolute)** `50`
Invoices addressed to the policyholder	**Document** Repair invoice › `customer_name`	**Form answer** `fullName`	**Person name**	**When several documents match**: **All must agree**

Rule name	Compare	With	Comparison	Settings
Bank account holder	**Document** Bank statement › `account_holder`	**Document** Identity card › `full_name`	**Person name**	**When the document is missing**: **Skip the rule**

A claim whose rules all match continues straight to payment; any `mismatch` or `inconclusive` rule sends it to the adjuster with the divergent documents marked. The video below (in Brazilian Portuguese) follows a claim checked by 14 such rules.



Sinistro, passo a passo (3/6): As 14 regras que conferem um sinistro automaticamente (in Brazilian Portuguese) Watch on YouTube

Bank: a mortgage application

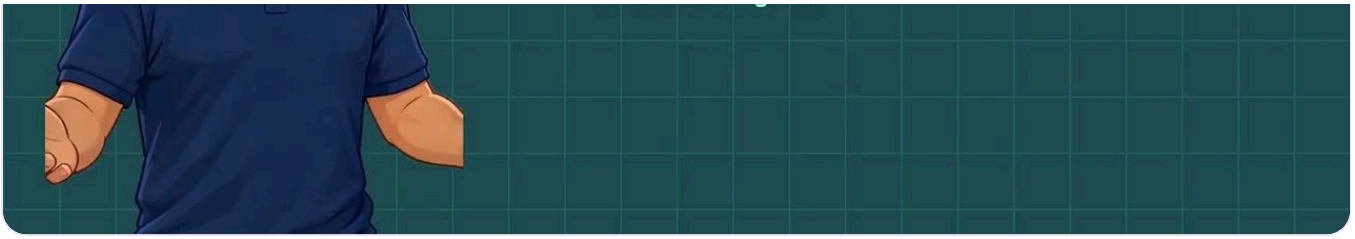
An applicant sends their last three payslips, their tax return, their identity card and the property's land registry record. The form asks for their declared monthly income and the property's district.

Rule name	Compare	With	Comparison	Settings
Income on every payslip	**Document** Payslip › `net_salary`	**Form answer** `monthlyIncome`	**Amount**	Tolerance `10` %; **When several documents match**: **All must agree**
Name on every payslip	**Document** Payslip › `employee_name`	**Document** Identity card › `full_name`	**Person name**	**When several documents match**: **All must agree**
Employer	**Document** Payslip › `employer`	**Document** Tax return › `employer`	**Text**	**Let AI settle close calls** on; **When**

Rule name	Compare	With	Comparison	Settings
				several documents match: **Only the first**
Property district	**Document** Land registry record › `distrito`	**Form answer** `propertyDistrict`	**Text**	—

The income rule shows why **When several documents match** matters: with **Any one is enough**, a single payslip close to the declared income would pass; with **All must agree**, every payslip has to support it.





Bank CAN on Cogneris: Mortgage documents to signed declaration in minutes Watch on YouTube

Reading the result

In all three cases the phase adds the same `crossCheck` block to the run (see What happens in a run): one `decision` for the case, and one entry per rule in `checks` with its verdict and the values compared. To route on it, add a **Conditional** on the field `crossCheck.decision`, or place a **Manual Review** after the phase, which never approves on its own a case whose cross-check asked for review.

Configuration

The panel shows the hint *Compare a field of one document against a field of another, or against what the customer typed in the form.*., one block per rule, **Add rule**, and **Let AI settle close calls**.

Field	What it does	Required	Format / values	Default	Valid mess
Description	Name shown on the node.	No	Text	*Cross-check*	—
Rule name	Names the rule for the reviewer and in the results.	No	Text	Empty	—
Compare › Source	Where the first value comes from.	Yes	**Document**, **Form answer**, **Process data**	**Document**	—

Field	What it does	Required	Format / values	Default	Valid mess
Compare › **Document**	The document role. Shown when **Source** is **Document**.	Yes, for **Document**	*Select a document…*, then the workflow's checklist slots (by title, or *Checklist slot #…* for a slot without a title), then your templates	*Select a document…* (none)	*Could load y temp You c still ty docu type field*
Compare › **Field**	The field to read. Hidden for **Face match**. When **Document** is a template, the field names of the template are suggested.	Yes, except for **Face match**	Field name or dotted path; for **Form answer** the answer's key; for **Process data** `identifier`, `requester.name` or `today`	Empty	—
With › **Source**, **Document**, **Field**	The second value, same options as **Compare**. For **Face match**, the selfie.	As for **Compare**	As for **Compare**	As for **Compare**	As for **Comp**

Field	What it does	Required	Format / values	Default	Valid mess
Comparison	How the two values are compared.	Yes	**Person name**, **Text**, **Exact**, **Document number**, **Date**, **Amount**, **AI judgement**, **Face match**	**Text**	—
Accept a surname changed by marriage	Tells the AI that a surname added or replaced by marriage is expected. Shown for **Person name**.	No	On / off	Off	—
Tolerance (%)	Allowed difference as a percentage of the larger amount. Shown for **Amount**.	No	Number, 0 or more	Empty (no tolerance)	—
Tolerance (absolute)	Allowed absolute difference, in the amount's own unit; wins over the percentage.	No	Number, 0 or more	Empty	—

Field	What it does	Required	Format / values	Default	Valid mess
	Shown for **Amount**.				
Instruction for the AI	The question for the AI. Shown for **AI judgement**; the box suggests *What should the AI judge about this pair?*	Yes, for **AI judgement**	Text	Empty	—
When several documents match	How the pairings of a rule are combined.	No	**Any one is enough**, **All must agree**, **Only the first**	**Any one is enough**	—
When the document is missing	What a document role that no document fills means.	No	**Send for review**, **Skip the rule**	**Send for review**	—
Let AI settle close calls	Lets the AI decide pairs the rule-based comparison cannot settle. Applies to every rule	No	On / off	On	—

Field	What it does	Required	Format / values	Default	Valid mess
	of the phase.				

Checklist slots are offered only once the workflow has been saved.

Buttons and actions

Button	What happens	Reversible?
Add rule	Adds an empty rule (**Text**, both sides **Document**). It is saved once you change one of its fields.	Yes — remove it
× (*Remove rule*)	Removes the rule. The phase's other settings, such as **Let AI settle close calls**, are kept. With no rule left the panel warns *Add at least one rule: a cross-check without rules fails the run.*	Add the rule again

Common problems

Message	Cause	What to do
Could not load your templates. You can still type a document type and a field name.	The template list did not load.	Reload the editor. The **Document** list keeps the checklist slots.
Cross-check step has no rules configured. Add at least one rule, or remove the step. (run error)	The phase has no rule.	Add at least one rule and publish again.
No documents found for cross-check. (run error)	The run reached the phase with no documents.	Check the phases before it and what the process was started with.

Message	Cause	What to do
A rule is always `inconclusive` with reason `role_not_present`	No document of the case fills the role — wrong template, or the document type differs from the one classified.	Check the role; use **Skip the rule** if the document is optional.
A rule is `inconclusive` with reason `missing_value`	The document does not carry the field, or the field name is wrong.	Check the field name against the extraction result.
A **Face match** rule is `inconclusive` with reason `face_needs_two_documents`	A side of the face rule is not a document.	Set both sides to **Document**.
Reasons starting with `rejected:` or `call_failed:`	The comparison service refused or failed the call.	The case goes to review; the reason names the cause.

Related

- Facematch — the dedicated face check.
- Manual Review
- Conditional
- Generate document — can print the cross-check decision and counts.
- Organize human review
- Design and publish a workflow

Last reviewed 2026-10-22 · Version 1.1.0 · Owner: product-documentation

Manual Review

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Stops the process and puts each of its documents in a review queue, until a person approves or rejects them. Optionally, it lets a process through without a person when every field the AI read is confident enough and nothing else needs a human eye.

Palette	**Nodes** › **Manual Review**
Outputs	One: once the review is approved, the run continues along every link of that output. A rejection does not take an output — see Approval and rejection. Before this version the editor also showed **Alternative**; a link still on it shows *Alternative not routed* and must move to the output before a new version is published.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

- After the phases whose results a person must confirm — typically **Extraction**, **Cross-check**, **Facematch** or a REST consultation.
- Before a phase that must only act on confirmed data, such as **Generate document** or **Send**.

The **Document intake with review**, **Identity verification** and **Business onboarding** workflow templates include this phase. Who reviews what is set up separately: see Organize human review.

What happens in a run

1. **The audit is gathered.** The phase collects the output of every phase completed before it (classification, extraction, cross-check, face match and so on), so the review screen can show everything the AI produced in one place.
2. **Every document gets a type.** A document that reached the review without a type (no **Classification**, **Zero Shot**, or **Extraction** with **Detect document type** ran before) is classified once, so that it lands in the right queue. An image that cannot be identified but contains a face is typed `selfie`. If classification fails, the document stays untyped and the review screen shows its file name.
3. **Fraud verdicts are read.** For documents checked by the fraud pass, the graded verdict is fetched, so that a suspect document lands in the fraud queue rather than in its type's queue.
4. **Automatic approval is tried**, only when **Skip review when every field is confident** is on — see Automatic approval.
5. **One review item per document is created**, with the status *pending*. Documents that failed in an earlier phase are left out. If every document failed earlier, the failed documents are put in review instead, so a person sees what happened; the review never passes on its own.
6. **The process waits** for the review, and its status in the Document Center becomes *awaiting review*.

A process with no documents at all completes the phase without waiting.

When the classification service refuses temporarily while checking an image for a face (step 2), the phase is run again after 30 seconds, then 2 minutes; after that it waits for review with that document untyped.

Review queues

Each document waits in exactly one queue:

Queue	Holds
One queue per document type	Documents of that type. The type is compared ignoring case and surrounding spaces.
Unidentified	Documents that reached review without an identified type.
Fraud review	Documents the fraud pass flagged.

Queue	Holds
Face conference	Selfies, for the face comparison.

The queues are not configured in the phase. An administrator assigns reviewers to them in **Administration › Review Queues**; a reviewer with no queue sees all of them. See Organize human review.

Automatic approval

With **Skip review when every field is confident** on, the process is approved without a person only when **all** of the following hold. If any does not, it goes to review as usual.

- It is not a correction round (a process resubmitted after a rejection).
- It has documents, and none failed in an earlier phase.
- No phase whose result only a person can judge completed before the review: **Facematch**, **Cross-check**, **Signature validation**, or a **REST integration** that publishes its answer for the review screen.
- An **Extraction** or **Zero Shot** phase ran before it.
- No document has a fraud signal or was blocked by the fraud pass.
- No document is a selfie, and every document has an identified type.
- Every field extracted on every document carries a confidence, and the lowest of them is at or above **Minimum confidence (%)**. A field the AI did not find counts as confidence 0.

An automatically approved process still records one review item per document, approved with no reviewer and a note giving the lowest confidence and the threshold. The step's output says it was approved automatically.

Approval and rejection

- **Approved** (every document decided and none rejected, or approved as a whole): the phase completes and the run continues to **every** phase connected to it, whichever output the connection was drawn from.
- **Rejected**: the run does not continue. A process from the Portal goes back to its requester; a process started inside Workflow ends as failed (`manual_review_rejected`). See Organize human review.

Configuration

Field	What it does	Required	Format / values	Default	Validation messages
Description	Name shown on the node.	No	Text	*Manual Review*	—
Skip review when every field is confident	Turns automatic approval on. The panel explains: *When every field the AI extracted on every document is at or above this confidence, the process is approved without a person. Fraud signals, selfies, unidentified documents, correction rounds and face match, KYB, cross-check or signature checks always go to review.*	No	On / off	Off	—
Minimum confidence (%)	The lowest field confidence that still allows automatic approval. Shown only when the option above is on.	Yes, when shown	Whole number from 1 to 100; other values are rounded and brought into range; an empty	95	—

Field	What it does	Required	Format / values	Default	Validation messages
			or invalid entry reverts to the previous value		

Settings the panel does not show are kept when you change these (for example keys added earlier by hand).

Common problems

Message	Cause	What to do
A process waits in review although **Skip review when every field is confident** is on	One of the conditions of Automatic approval was not met — most often a field below the threshold, a field without confidence, a selfie, or a cross-check or facematch before the review.	Expected. Lower the threshold only if the risk is acceptable.
A phase connected to **Alternative** runs after an approval	A version published earlier keeps running every link it has.	Connect the link to the output and publish again; a rejection stops the run, so it cannot take a path of its own.
A document waits in *Unidentified*	No phase gave it a type and the classification at review time did not identify it.	Add a **Classification** or **Zero Shot** phase, or let a reviewer classify it.

Related

- Organize human review — queues, reviewers, returning a process to its requester.
- Cross-check, Facematch, Signature validation — their results always need a person.

- Extraction, Zero Shot — provide the field confidences.
- Design and publish a workflow

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Conditional

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

A *Conditional* phase asks a yes/no question about the result that reached it and sends the process down its **True** or its **False** output. It changes no document; it only chooses the branch.

Palette	**Nodes** › **Conditional**
Outputs	**True** — the conditions are met. **False** — they are not, or they could not be evaluated. The output not taken is skipped.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

- Deliver straight away when a check passed, and send everything else to a person: for example **True** → *Send*, **False** → *Manual Review*.
- Branch on what an earlier phase decided: the face check, the cross-check, the classification, or the answer of a *REST integration*.

A *Conditional* accepts only **one incoming link**.

What happens in a run

1. **What it reads.** The phase reads the output of the phase that led into it. If there is none, it reads what the process was started with.
2. **How a field is found.** Each condition names a **Field** as a path: property names separated by dots, and a position in a list in square brackets, counted from 0 — for

example `results[0].classification.documentType`. Names are case-sensitive. The path is looked for first at the top of the result and then, if it is not there, inside each wrapped earlier result (up to 8 levels). This is what lets a condition placed after a *Manual Review* still read what the phases before the review produced. Where the path exists at the top, the top wins.

3. **How each condition is decided.** The **Operator** compares the value found at the path with the **Value** you typed — see Operators.
4. **How the answer is decided.** With **All conditions (AND)** every condition must be true; with **Any condition (OR)** one is enough. True → the **True** output; otherwise → **False**.
5. **What it passes on.** The output is the result it received, unchanged, plus three entries: `branch` (`true` or `false`), `decision` (the same as a true/false value) and `conditionalEvaluated` — one line per condition with the field, the operator, the value, the **value actually read** (empty when the field was missing) and the result. This is where to look when a process took the branch you did not expect.
6. **Routing.** Only the links leaving the chosen output stay live. Phases reachable only through the other output are marked skipped. A link that leaves the phase without an output name is never followed.

The phase never fails the run. Anything it cannot evaluate routes to **False**:

Situation	Result
No condition with a **Field** (or no configuration at all)	The whole phase answers **False**.
A workflow saved before conditions were a list (one **Field**, **Operator** and **Value** stored on their own)	Read as a single condition, exactly as the editor shows it when the phase is opened.
The field does not exist in the result	That condition follows the "missing field" column of Operators.
The result is not valid JSON	Every field counts as missing.
An operator the service does not know (only possible in a configuration not made in this panel)	That condition is false.

Fields you can compare

The **Field** box accepts any path. Its suggestion list offers the fields of an imported schema and the answer of every *REST integration* placed **before** this phase. The paths below are written by the other phases; each exists only if that phase ran before the *Conditional* on the same route.

Path	Written by	Values
`results[0].classification.documentType`	Classification, or Extraction with **Detect document type**	The document type, as the classifier names it
`results[0].classification.confidence`	Same	The classifier's confidence, as a number
`results[0].identified`	Extraction with **Detect document type**	`true`, or `false` when the type is `UNKNOWN` or empty
`results[0].zeroShot.documentType`	Zero Shot	The document type the zero-shot reading returned
`results[0].fields.metadata.…`	Extraction	An extracted field, under the name the extraction gave it
`results[0].fields.fraudBlocked`	The extraction, when the service reports it	`true` / `false`

Path	Written by	Values
`results[0].facematch.score`	Facematch	The similarity score of the first document
`facematch.decision`	Facematch, or a *Cross-check* rule with the face comparator	`pass`, `review` or `notApplicable` (no document was face-checked)
`crossCheck.decision`	Cross-check	`pass` or `review`
`restCall.….ok`	REST integration — the part after `restCall.` is that phase's node id	`true` / `false`
`restCall.….found`	Same	`false` when the lookup found nothing
`restCall.….status`	Same	The HTTP status the system answered
`restCall.….data.…`	Same	A value of the mapped answer

`results[0]` is always the **first document** of the process. To gate the whole process on a face check or a cross-check, use `facematch.decision` or `crossCheck.decision`, which summarize every document.

Operators

The **Operator** list shows symbols. The **Value** box appears only for operators that compare with a value.

Symbol	Operator	True when	Value	Missing field
=	equals	The value read equals **Value**	Required	false
≠	not equals	The value read differs from **Value**	Required	**true**
>	greater than	The value read is a number greater than **Value**	A number	false
≥	greater or equal	… greater than or equal to **Value**	A number	false
less-than sign	less than	… less than **Value**	A number	false
≤	less or equal	… less than or equal to **Value**	A number	false
∈	in	The value read is one of the items of **Value**, a comma-separated list (`passport, id_card`)	A list	false
∉	not in	The value read is none of the items of **Value**	A list	false
⊃	contains	The value read is text that contains **Value**, or a list with an item equal to **Value**	Required	false
?	exists	The field exists and is not null	None	false
∅	is empty	The field is missing, null, empty text, an empty list or an empty object	None	**true**

How values are compared:

- **Value** is always stored as text. Equality is therefore a **text comparison that ignores upper and lower case**: `PASSPORT` equals `passport`, and a true/false field equals `true` or `false`. A number equals **Value** only if it is written the same way (`0.85` equals `0.85`, not `0.850`).
- The four number operators read both sides as numbers, with a **dot** as the decimal separator. If either side is not a number, the condition is false.
- **in** and **not in** split **Value** at the commas and trim the spaces; the comparison ignores case. When the field read is itself a list, they check whether that list holds **Value** as a whole (no splitting).
- **contains** on a number or a true/false value is always false.
- An empty **Value** is not stored.

Configuration

Select the phase to open **Phase Configuration**. Its header shows the phase type as `CONDITIONAL`.

Field	What it does	Required	Format / values	Defa
Description	The name shown on the node	No	Text	*Conditi*
Match	Whether every condition or any one condition must be true	Yes	**All conditions (AND)** / **Any condition (OR)**	**All condit (AND)**
Import JSON schema	Reads a `.json` file and adds its fields to the **Field** suggestions. Accepts a JSON schema (its `properties`, with `title` as label), a list of fields (`fields` with `key`), or a sample object (its top-level names). Each name is turned into	No	A JSON file	—

Field	What it does	Required	Format / values	Defa
	`results[0].fields.metadata.` followed by the name, unless it already starts with `results[`. The list is stored with the phase.			
Field (per condition)	The path to read — see Fields you can compare. Suggestions: imported fields, then the answers of earlier *REST integration* phases, labelled *Query: …*	Yes — a condition without one is dropped when saved	A path	Empty
Operator (per condition)	How to compare — see Operators	Yes	One of the 11 symbols	=
Value (per condition)	What to compare with. Hidden for **exists** and **is empty**, and removed from the condition when you switch to them.	For the 9 comparing operators	Text; a number for the number operators; a comma-separated list for **in** / **not in**	Empty

Changes go into the workflow draft. Save it with **Save Draft**; processes use them only after you publish.

Buttons and actions

Button	What happens	Reversible?
Add condition	Adds an empty condition row. It is stored only once its **Field** is filled.	Yes — remove it
× (**Remove condition**)	Removes the condition at once, without confirmation.	Use **Undo** in the editor toolbar
Import JSON schema	Replaces the stored list of suggested fields with the file's. Conditions already written are not changed.	Import another file

Common problems

Message	Cause	What to do
Could not read that file as JSON.	The imported file is not valid JSON	Export the schema again as JSON.
No fields were found.	The file is valid JSON but has no field names in a shape the editor reads	Use a JSON schema with `properties`, a `fields` list, or a sample object.
This field reads query …, which no longer exists or is not before this conditional.	The **Field** reads a *REST integration* that was deleted, disconnected or moved after this phase. The service would find nothing and route **False**.	Pick the field again from the suggestions, or move the *REST integration* before this phase. The editor does not stop you publishing.
The process always takes **False**	The path is not in the result (a typo, wrong case, a phase that did not run on this route), the value is compared as text, or no condition has a **Field**	Open the run's `conditionalEvaluated`: the value actually read shows which. Check the path's case and the decimal separator.

Message	Cause	What to do
A phase after the **False** output is shown as skipped	Expected: the process took **True**	—

The editor shows no publication message specific to this phase.

Related

- Design and publish a workflow
- REST integration — the `restCall` answers a condition can read
- Manual Review — the usual destination of **False**
- Workflow administration screens

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

AI Executor

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

AI Executor is **no longer offered in the palette**: the Workflow service has no step that runs it, and a workflow that contains it **cannot be published**. A draft that still has one opens with the notice *This phase cannot run yet* on the node and in its settings; remove the node before you publish.

Palette	Not offered (no longer)
Outputs	None is ever taken, because the phase never completes
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

Not yet. No workflow template adds it; it appears only if you add it from the palette or the guided **Step type** list.

For AI work on documents, use the phases the service runs: Classification, Extraction, Zero Shot, Facematch and Cross-check (whose *semantic* comparator takes a prompt).

What happens in a run

When a process reaches *AI Executor*, the service finds no step for the phase type `ai_executor` . The step fails with the error *No executor registered for phase 'ai_executor'* and **the whole run fails**. No AI model is called and no answer is produced.

Publishing does **not** warn you: the service accepts a workflow containing this phase.

Prompt, model and answer

There are none. The panel has no prompt field and no model choice, and since the phase never runs, nothing is done with an answer. Whatever you type in **Config (JSON)** is stored with the phase and ignored; text that is not valid JSON is not saved.

Configuration

The panel has no form of its own. Its header shows the phase type as `AI_EXECUTOR`.

Field	What it does	Required	Format / values	Default	Validation messages
Description	The name shown on the node	No	Text	*AI Executor*	—
Config (JSON)	JSON stored as the phase configuration. Not used by the service.	No	Valid JSON, or empty	Empty	*This is not valid JSON, so it was not saved. Fix it or clear the field.*

Common problems

Message	Cause	What to do
The run fails at *AI Executor* with *No executor registered for phase 'ai_executor'*	The service has no step for this phase	Remove the phase (reconnect the phases around it), publish, and start the process again.

Related

- Cross-check — AI comparison with a prompt
- Zero Shot
- Design and publish a workflow

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Send Message

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Send Message is **no longer offered in the palette**: the Workflow service has no step that runs it, and a workflow that contains it **cannot be published**. A draft that still has one opens with the notice *This phase cannot run yet* on the node and in its settings; remove the node before you publish. To deliver a result, use Send.

Palette	Not offered (no longer)
Outputs	None is ever taken, because the phase never completes
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

Not yet. The templates **Business onboarding** and **Contract signature** of **Create Workflow** used to end with it; they now end with a Send phase. A draft created from them before still has *Send Message*: replace it with **Send** (or remove it) before you publish.

What happens in a run

When a process reaches *Send Message*, the service finds no step for the phase type `send_message` . The step fails with the error *No executor registered for phase 'send_message'* and **the whole run fails**. Nothing is sent: no message, no e-mail and no webhook call.

Publishing does **not** warn you: the service accepts a workflow containing this phase. Unlike *Send*, it does not even check that the endpoint is HTTPS. The only thing publishing does for it

is create a signing secret if an endpoint is set and none exists yet.

Channels, recipients and message templates

There are none. The panel has no channel (e-mail, SMS, WhatsApp), no recipient and no message text or placeholders. To notify a person through a channel, see the Signature phase, which has its own channel setting, or deliver to your system with Send and send the message from there.

Configuration

The panel is the same as the *Send* panel. The fields are stored with the phase but are not used, because the phase never runs. Its header shows the phase type as `SEND_MESSAGE`.

Field	What it does	Required	Format / values	Default	Validation messages
Description	The name shown on the node	No	Text	*Send Message*	—
Subject	Stored; not used	No	Text	Empty	—
Integration name	Stored; not used	No	Text	Empty	—
Endpoint URL (HTTPS)	Stored; not used. Setting it creates a **Signing secret**, as on *Send*.	No	An `https://` URL or a test URL	Empty	*Enter a valid HTTPS URL.* (in the panel only)
HTTP method	Stored; not used	No	POST, PUT, PATCH, DELETE	POST	—

Field	What it does	Required	Format / values	Default	Validation messages
Description (optional)	Stored; not used	No	Text	Empty	—
Signing secret	Shown to administrators only, while an endpoint is set. Not used.	—	64 hexadecimal characters	Created with the endpoint, or at publish	—

Buttons and actions

The panel shows the same buttons as *Send* — **Test endpoint**, **Copy**, **Regenerate**, **Generate test URL**, **Use as endpoint**, **View requests** and **Revoke** — and they behave as described in Send. **Test endpoint** really calls the endpoint. A test URL generated here receives nothing, because the phase never sends.

Common problems

Message	Cause	What to do
The run fails at *Send Message* with *No executor registered for phase 'send_message'*	The service has no step for this phase	Replace the phase with Send, publish, and start the process again.
Enter a valid HTTPS URL.	The endpoint is not an absolute `https://` address	Not relevant while the phase cannot run.
No requests received yet. Run the workflow to see deliveries here. in **View requests**	The phase never sends	Use a test URL on a Send phase.

Related

- Send — the phase that delivers results

- Signature
- Design and publish a workflow

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Generate document

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Renders a PDF from one of your published document templates, filled with the data of the case, and adds it to the process's documents and to the Document Center.

Palette	**Nodes** › **Generate document**
Outputs	One output.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

- To produce the document a customer signs or receives at the end of a case — a proposal, a contract, a declaration.
- Before **Signature**: the generated PDF is the document **Signature** sends by default.
- After **Manual Review** when the document must only contain confirmed data.

The **Contract signature** workflow template uses it: **Extraction**, **Generate document**, **Signature**, **Send**.

What happens in a run

1. **The case is sent to the template service.** The phase does not decide what each placeholder of the template holds. It sends the whole case, and the **template's own bindings** pick the values:

Part of the case	What it contains
Documents	For each document: name, document type, template, checklist slot, whether it failed, and its extracted fields (by full dotted path and by short name).
Form	The form answers, by dotted key.
Process	The process data (for example `identifier`, `requester.name`, `today`).
Cross-check	When a **Cross-check** ran before: `decision`, the counts `summary.total`, `summary.match`, `summary.mismatch`, `summary.inconclusive`, `summary.notApplicable`, and `checksCount` (how many checks ran — not how many failed).

How a binding addresses these values, and what a binding does when a value is missing, is set on the template, not in the workflow. The phase works after a **Manual Review** too: it still finds the documents and fields of the phases before the review.

2. **If a value that matters cannot be filled**, the template service answers *needs review*:
   - with **Print even with gaps** off (default), no PDF is produced; the run stops at this phase and waits for a person, with the list of missing keys and why each is missing (how such a run is resumed is decided by the service and not shown in the editor);
   - with it on, the PDF is produced with the gaps.
3. **The PDF is stored** under **File name** (or the name the template service suggests), registered as a document of the run with the type `generated-document`, and filed in the Document Center beside the case's other documents. If the Document Center refuses it, the run continues and the refusal is recorded.

The phase adds a `generateDocument` block to the run's data: `status` (`rendered` or `needs_review`), `templateCode`, the template `versionNumber`, a `renderId`, the `document` (name, location, content type, size, and `ecmDocumentId` — the Document Center id, the one an integration should keep), whether it reached the Document Center, and every `missing` key (key, reason, whether it was blocking).

Retries. When the template service or storage is temporarily unavailable, the phase is run again up to 3 times. A *needs review* answer is never retried, since it would consume the render quota again for the same answer.

Configuration

Field	What it does	Required	Format / values	Default	Validation messages
Description	Name shown on the node.	No	Text	*Generate document*	—
Template code	The published document template this phase renders. It is the template's stable code (slug), not its display name — for example `aprovacao-credito-habitacao`.	Yes	Text	Empty	None in the panel; an empty code fails the run (see below).
File name	The name of the generated PDF. `.pdf` is added when missing.	No	Text	Empty — the name the template service suggests	—
Print even with gaps	Produce the PDF even when a value that matters cannot be filled. Keep it off for anything a	No	On / off	Off	—

Field	What it does	Required	Format / values	Default	Validation messages
	person signs: a blank where a name belongs looks finished.				

The template, its bindings and what counts as a value that matters are managed in the template service, not in this panel.

Common problems

Message	Cause	What to do
generate_document needs a templateCode in its node configuration. (run error)	**Template code** is empty.	Fill it in and publish again.
template-be refused to render '…': … or *template-be could not render '…': …* (run error)	The template service refused the request — for example an unknown or unpublished template code.	Check the template code against the template service.
The run waits at this phase with `status` `needs_review`	A value that matters could not be filled; the `missing` list names it.	Fix the data or the template's binding.
`ecmLanded` is false, with an `ecmLandingError`	The Document Center refused the PDF.	The PDF is still on the run; contact support if it repeats.

Related

- Signature — signs the generated document.
- Cross-check — its decision and counts can be printed.
- Manual Review

- Send — delivers the document.

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Signature

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Sends a document produced earlier in the run to the customer for electronic signature, waits for the answer, and replaces the draft with the signed copy.

Palette	**Nodes** › **Signature**
Outputs	One output, taken once the document is signed. A refused, expired or cancelled request fails the run.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

After **Generate document** and before **Send**, so the delivery carries the signed copy and not the draft. The panel says so: *Parks the run and asks the customer to sign what an earlier node produced. It belongs after Generate document and before Send, so the webhook carries the signed copy and not the draft.*

The **Contract signature** workflow template uses it.

Each Signature phase asks **one** signer. For several signers in sequence (for example the insured, then the insurer), add one Signature phase per signer. Signer profiles are not used here: they belong to Signature validation, which checks signatures already present in a PDF.

What happens in a run

First passage — the request.

1. The document to sign is the most recent non-failed document of the run whose type is **Document to sign** (`generated-document` by default, which is what **Generate document** produces).
2. The signer's name, and their email or phone depending on **Delivery channel**, are read from the case: the key is looked up in the form answers (a leading `formAnswers.` is ignored), then in the process data.
3. The signing service creates the request — the *envelope* — named **Envelope name** (or the document's file name), with the deadline, reminder, jurisdiction and **Case name**, and sends the signing link and the one-time code over the channel.
4. The run waits. Creating the same request twice is impossible: a repeated attempt for the same run and phase finds the envelope already raised instead of asking the customer again.

While it waits, the service checks the signing service regularly and wakes the phase when the request has an answer.

Second passage — the answer.

Answer	Result
Signed	The signed PDF is stored as `…-assinado.pdf` and **replaces** the draft as the run's document, so later phases (for example **Send**) deliver the signed copy. It is filed in the Document Center as a new version of the generated document, so the same document id now resolves to the signed copy and the draft stays in its history. The signing audit trail is filed beside it as `…-trilha-de-auditoria.pdf`. The run continues.
Refused by the signer	The run fails with `signature_refused`.
Not signed before the deadline	The run fails with `signature_expired`.
Cancelled	The run fails with `signature_cancelled`.
Still open	The run keeps waiting.

A signature is billed once, when the signed document is collected. Raising a request, or a request that is refused or expires, is not billed. If the Document Center refuses the signed version or the audit trail, the run still continues and the refusal is recorded.

The phase adds a `signature` block to the run's data: while waiting, `status` `awaiting_signature`, the envelope id, the channel, the deadline and whether the message was delivered (with the reason when it was not); once signed, `status` `signed`, the signing time, the signed `document` (including `ecmDocumentId`), the sealed document's fingerprint and the audit-trail filing.

Retries. When the signing service or storage is temporarily unavailable, the phase is run again up to 3 times. A refusal or an expiry is an answer and is never retried.

Configuration

Field	What it does	Required	Format / values
Description	Name shown on the node.	No	Text
Delivery channel	How the signing link and the one-time code reach the signer.	Yes	**Email**, **SMS**, **WhatsApp**
Signer name field	Where the signer's name is read from.	Yes	A form-answer or process key, for example `formAnswers.nome_completo`
Signer email field	Where the signer's email is read from. Shown when the channel is **Email**.	Yes, for **Email**	Key, for example `formAnswers.email`

Field	What it does	Required	Format / values
Signer phone field	Where the signer's phone is read from. Shown when the channel is **SMS** or **WhatsApp**. The phone must carry the country code.	Yes, for **SMS** and **WhatsApp**	Key, for example `formAnswers.telefone`
Expires after (days)	Days the signer has before the request lapses and the run fails with `signature_expired`. Leave it blank for no deadline.	No	Whole number, 1 or more
Remind after (days)	Days after the request to send the same link again to a signer who has not answered.	No	Whole number, 1 or more
Jurisdiction	The legal regime the signed package declares.	No	**Environment default**, **Brazil (MP 2.200-2 / Law 14.063)**, **European Union (eIDAS)**
Document to sign	The document type of the run document to sign. Change it only if another phase produces the document.	Yes	Document type
Envelope name	What the signer is told they are signing.	No	Text, for example *Credit agreement*

Field	What it does	Required	Format / values
Case name	Names the case in the signing app, so a signer with several requests can tell them apart.	No	Text, for example *Home loan*

The panel always saves the name, email and phone keys, the channel and **Document to sign**: leaving one of them empty saves its default. **Expires after (days)** is different: a new phase starts at 7, and emptying it saves *no deadline*, so the request does not expire.

Every field shows its explanation under it, and the key fields suggest an example (`formAnswers.nome_completo`, `formAnswers.email`, `formAnswers.telefone`, `generated-document`).

The jurisdiction also sets the language of the signing request. The service refuses a jurisdiction other than Brazil or EU.

Common problems

Message	Cause	What to do
signature found no '...' document on this run to sign. The signature node must come after a node that produces one (generate_document). (run error)	No document of the run has the type in **Document to sign**.	Place **Generate document** before, or correct **Document to sign**.
signature could not resolve the signer's name from '...'. The key must name a field the case actually carries, or the node must set signerName. (run error)	**Signer name field** names a key the case does not have, or it is empty.	Use the key of the form answer that holds the name.

Message	Cause	What to do
signature is configured to deliver by email but '…' resolved to nothing. (run error)	No email for the signer.	Correct **Signer email field**, or make the form field required.
signature is configured to deliver by … but '…' resolved to nothing. (run error)	No phone for SMS or WhatsApp.	Correct **Signer phone field**.
sign-be refused the signature request: … (run error)	The signing service refused the request.	Read the detail; check the signer's email or phone format.
`signature_refused`, `signature_expired`, `signature_cancelled` (run error)	The signer declined, did not sign in time, or the request was cancelled.	Start a new process if the document must still be signed.
`delivered` is false in the phase output	The message did not reach the signer (for example a bounced email). The request exists and the run waits.	Check the signer's contact details.

Related

- Generate document — produces the document to sign.
- Signature validation — checks digital signatures already in a PDF.
- Send — delivers the signed copy.
- Design and publish a workflow

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Signature validation

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Verifies the digital (ICP-Brasil) signatures already present in the process's PDF documents against a signer profile and a rule, and sends the process down **OK** or **NOK**.

Palette	**Nodes** › **Signature validation**
Outputs	**OK** — every checked document satisfies the rule. **NOK** — at least one does not. Both must be connected before the workflow can be published. A technical failure takes neither: the run fails.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

When a process receives PDFs that must already be signed by specific people or companies — identified by the CPF or CNPJ in their certificate — and you must route on whether they are. Register who may sign first, in a **signer profile**: see Validate digital signatures.

To collect a new signature from the customer, use Signature instead.

What happens in a run

Which documents. Every PDF the process received that did not fail in an earlier phase. Documents the run produced itself (for example a generated document) are not checked. Each checked document must satisfy the rule on its own.

How each document is judged.

1. Every signature in the document is verified: integrity, a trusted certificate chain, the certificate valid at signing time, and not revoked.

2. Each signature is matched to the profile by the CPF or CNPJ of the certificate's **subject** — never by name. When a signer of the profile lists allowed issuing authorities, the certificate's issuer must be one of them.

3. The document is **OK** only when:

   - it has at least one signature;
   - **every** signature is valid **and** belongs to a signer of the profile — a single signature by someone outside the profile makes the document **NOK**, whatever the rule;
   - the rule is met:

Rule	Met when
Any listed signer	At least one signer of the profile signed validly.
All required signers	Every signer marked required in the profile signed validly.
N of M	At least **Minimum number of signers** different signers of the profile signed validly.

4. The process takes **OK** when every checked document is OK, and **NOK** otherwise — including when there is no PDF to check (`no_signature`).

Reasons. A **NOK** carries the reasons of every document:

Reason	Shown as
`no_signature`	*The document has no signature*
`invalid_signature`	*The signature is not valid*
`document_modified`	*The document was modified after signing*
`certificate_untrusted`	*The certificate is not from a trusted authority*

Reason	Shown as
`certificate_expired_at_signing`	*The certificate had expired at signing time*
`certificate_revoked`	*The certificate was revoked*
`signer_not_allowed`	*The signer is not in the profile*
`required_signer_missing`	*A required signer did not sign* (rules **All required signers** and **N of M**)
`indeterminate`	*Revocation could not be checked*

When revocation cannot be checked. If the only thing standing between a document and **OK** is a revocation service that did not answer, the phase does not decide yet: it is run again after 30 seconds, 2 minutes, 8 minutes, 30 minutes and 30 minutes (5 retries by default). If the answer is still missing after the last one, the process takes **NOK** with `indeterminate`. A revoked certificate is a verdict and takes **NOK** immediately.

Technical failures fail the run — they never become a **NOK**: the profile is missing or empty, the rule does not fit the profile, the signature verifier or its trusted authorities are unavailable, the signers' identifiers cannot be decrypted, or a document is unreadable, encrypted or too large. A document that cannot be read from storage is retried.

Outputs. A `signatureValidation` block in the run's data, keyed by the phase's node, so several Signature validation phases keep their own results: `ok`, the rule, the profile, one entry per signature (signer, masked identifier, issuer, signing time, whether the time is a trusted timestamp, status *Valid*, *Invalid* or *Indeterminate*, reasons, revocation check), one entry per document, the reasons, the trust-store version and the duration. A JSON validation report is also filed in the Document Center beside the case (`reportRef`); if that fails, the run continues and `reportError` says why. Identifiers appear only masked.

Configuration

The panel has two tabs, **Signers** and **Test**.

Signers

Field	What it does	Required	Format / values	Default	Validation messages
Description	Name shown on the node.	No	Text	*Signature validation*	—
Signer profile	The company's list of allowed signers to match against. Each option reads *… (… signers)*. *Signers are registered per tenant, in a profile.* **Manage profiles** opens **Signer profiles**. Once chosen, the panel shows *… signers in the profile, … of them required.*	Yes	One of your signer profiles	*Choose a profile* (none)	*Choose a signer profile.* · *The selected profile has no signers.* · *Could not load the signer profiles.*
Rule over the signatures	Which signatures make a document OK.	Yes	**Any listed signer** (*At least one valid signature from a signer in*	**Any listed signer**	*The profile has no required signer, so this rule can never pass.*

Field	What it does	Required	Format / values	Default	Validation messages
			the profile.), **All required signers** (*Every signer marked as required must have signed.*), **N of M** (*At least N different signers from the profile must have signed.*)		
Minimum number of signers (out of …)	N, for **N of M**. Shown only for that rule.	Yes, for **N of M**	Whole number from 1 to the number of signers in the profile	1 when you choose **N of M**	*The minimum must be at least one signer.* · *The minimum cannot exceed the … signers in the profile.*

Below the fields: *Connect both outputs: OK continues the flow and NOK handles a rejected document. Publishing is refused while either one is unconnected.*

The phase stores only the profile reference and the rule; the signers themselves stay in the profile, so editing the profile changes what every workflow using it accepts.

Test

Check a sample PDF against this profile and rule. Nothing is recorded in any run.

Field	What it does	Required	Format / values	Default	Validation messages
PDF document	The sample document to check.	Yes	A PDF file	None	*Choose a signer profile first.* (no profile chosen yet)

The result is the **Signature validation report**: **Result**, the reasons, and one line per signature with **Signer** (*Issued by …*), **Identifier**, **Signed at** (*Trusted timestamp* or *Declared time, not proven*) and **Status**. With no signature: *No signature was found in the document.* When the report was filed: *Stored validation report*.

Buttons and actions

Button	What happens	Reversible?
Manage profiles	Opens **Signer profiles**.	Yes
Validate document	Runs the chosen PDF through the same check a run makes (*Validating…*) and shows the report. Enabled once a PDF and a profile are chosen. Nothing is recorded in any run.	Yes

Common problems

Message	Cause	What to do
Signature validation "…": the … output is not connected. (publish)	**OK** or **NOK** leads nowhere.	Connect both outputs.

Message	Cause	What to do
Signature validation "…": choose an existing signer profile. (publish)	No profile, or the profile no longer exists.	Choose a profile.
Signature validation "…": the signer profile has no signers. (publish)	The profile is empty.	Add signers to the profile.
Signature validation "…": the rule does not fit the signer profile. (publish)	**N of M** with N out of range, or **All required signers** on a profile with no required signer.	Fix the rule or the profile.
The document could not be validated. (test)	The test failed.	Check the file is a readable PDF; try again.
Too many tests in a short time. Wait a moment and try again. (test)	Too many tests (10 per minute by default).	Wait and test again.
A process takes **NOK** with *The signer is not in the profile*	A document carries a signature by someone outside the profile — every signature must belong to the profile.	Add that signer to the profile, if they are allowed.
Processes take **NOK** with *Revocation could not be checked*	The revocation service did not answer during all retries.	Restart the process later.

Related

- Validate digital signatures — signer profiles step by step.
- Signature — collect a signature from the customer.
- Manual Review — a manual review never approves on its own after a signature validation.
- Design and publish a workflow

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

REST integration

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Calls one of your systems' REST APIs through a **connection**, builds the request from the data of the process, classifies the answer and routes the process through **Success**, **Not found** or **Error**. The answer can be kept as a fact of the process and certified with a PDF.

Palette	**Nodes** › **REST integration**. Connector presets, such as the VIES check, are added from **Pre-configured** as a REST integration with its settings filled in.
Outputs	**Success** — the call (or a check before the call) produced an answer. **Not found** — the destination said there is nothing for this lookup. **Error** — a permanent failure, or a transient failure whose retries ran out. See What happens in a run.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

The REST integration has more settings than any other phase, so its reference is split into several pages:

Page	What it covers
Overview and VIES tutorial	When to use it, **Connections**, the tabs at a glance, checks before the call, and a worked example with the VIES preset
Request	The **Request** tab: connection, method, path and path parameters, query parameters, headers, timeout, retries, success status codes

Page	What it covers
Schemas and mapping	The **Schemas**, **Request mapping** and **Response mapping** tabs, sources, and every transform
Result, Test and Publication	The **Result**, **Test** and **Publication** tabs
Messages	Every message the REST integration can show, with cause and fix

When to use it

- To look something up in a system of yours or of a public body — a customer record, a tax number, a registry — and branch on the answer.
- To push data from the process into a system of yours (create or update a record) and continue only when it accepted it.
- To keep the answer of an external consultation as evidence of the process, with a PDF certificate.

To send the **final** result of a workflow to your system, the **Send** phase with a webhook is usually simpler — see Connect your systems.

What happens in a run

1. **The connection is resolved.** The phase reads its connection — base URL, authentication, default headers, timeout. A deleted connection, or one of another company, is an error of kind `connection`.

2. **The request is built** from the input of the phase (the output of the previous phase) and, where the mapping asks for them, from the process data: form answers (`$form`), process metadata (`$process`), documents by type (`$doc`) and run metadata (`$run`). For POST, PUT and PATCH the **Request mapping** builds the JSON body; GET and DELETE send no body. Path and query parameters are filled in the same way.

3. **Checks before the call** (the *Checks before the call* of the **Result** tab) are evaluated on the request. The first one that matches decides without calling the destination.

4. **The request is validated** against the **Request schema**, when there is one.

5. **The call is made** once, with the connection's credentials applied last, so nothing in the phase can replace them. The timeout is the phase's **Timeout (seconds)**, capped by the connection's. With **Send an Idempotency-Key that stays the same across retries** on, every attempt of the same run and phase carries the same `Idempotency-Key`; a correction round uses a new one. Redirects are not followed. For OAuth2 connections a rejected token (HTTP 401) is refreshed once, without counting as an attempt.

6. **The answer is classified:**

Answer	Result
A **Success status codes** status, and no rule of **Rules on the response body** matched	Success. The body, translated by the **Response mapping** (or the raw body without rules), becomes the answer.
A rule of **Rules on the response body** matched	What the rule says: Success, Not found, Transient (retried) or Permanent (error)
A status listed in **Not-found status codes (4xx)** (404 by default)	Not found
5xx, 408, 429, no answer within the timeout, a host that cannot be resolved, a connection failure	Transient: retried
Any other status, a redirect, a body larger than 5 MB, a success body that is not JSON or does not match the **Response schema**	Permanent: error

7. **Retries.** A transient failure is retried up to **Retries** times. The process waits **Wait between retries (seconds)**, or longer when the destination asks for it with `Retry-After`, never more than 600 seconds. No worker is kept busy while it waits.

8. **Routing.**

   - Success → **Success**.

- Not found → **Not found** when that output is connected. When it is not, the result goes to **Error**, never to **Success**.
- Permanent failure or exhausted retries → **Error** when it is connected. When **Error** is not connected either, the step fails and the process fails with it.

9. **Publication.** On the outcomes chosen in the **Publication** tab, the mapped answer is published as a fact of the process under its document type, and a PDF certificate is stored with the process documents. A later execution that no longer qualifies withdraws the fact and the certificate published before.

The phase adds its result to the data the next phase receives, under `restCall` and the phase's node id, keeping everything it received:

Path	Content
`restCall.` node id `.ok`	`true` for Success and Not found, `false` for Error
`.found`	`false` for a not-found result, `true` for every other success
`.status`	The HTTP status, when the destination was called
`.data`	The mapped answer, on Success only
`.attempts`, `.durationMs`	How many attempts were made, and how long the last one took
`.error.kind`	On Error: `transient`, `permanent`, `request_schema`, `response_schema`, `ssrf`, `connection`, `not_found` or `preflight`
`.error.message`, `.error.status`, `.error.bodyExcerpt`	On Error: the explanation, the status and up to 2 KB of the response body, with the connection's secrets removed

A **Conditional** placed after the phase offers these paths in its field picker.

Configuration

The panel shows **Description** (the name of the phase on the canvas) and then seven tabs, in this order. Each field is documented on the page linked.

Tab	What it holds	Page
Request	**Connection**, **Method**, **Path**, **Path parameters**, **Query parameters**, **Headers**, **Timeout (seconds)**, **Retries**, **Wait between retries (seconds)**, the Idempotency-Key option, **Success status codes**	Request
Schemas	**Request schema**, **Response schema**, **Input example**	Schemas and mapping
Request mapping	How the request body is built	Schemas and mapping
Response mapping	How the answer is translated into `data`	Schemas and mapping
Result	**Not-found status codes (4xx)**, **Rules on the response body**, **Checks before the call**	Result, Test and Publication
Publication	**Publish as a process fact**, **Proof of the query**	Result, Test and Publication
Test	One real call with the current settings	Result, Test and Publication

Above the tabs, a list shows every problem the editor finds in the settings while you edit — see Messages. A phase added from a preset also shows a **preset** banner — see the VIES tutorial in Overview and VIES tutorial.

A workflow can have at most 20 REST integration phases; publication refuses more.

Buttons and actions

Button	What happens	Reversible?
Manage connections	Opens **Connections**	—
Load from a run	Loads up to five recent outputs of the phase that feeds this one, to use as **Input example**	Yes
Test the call › Call now	Makes one real call to the destination	**No** — the destination receives the call
Preset parameters › Apply parameters	Replaces the whole configuration of a preset phase with the preset's, for the new parameters; the phase keeps its links to other phases	Only by editing again

Common problems

Message	Cause	What to do
Choose a connection.	No connection selected	Select one in **Connection**, or create it in **Connections**.
The process fails at this phase	An error with **Error** unconnected, or a not-found result with neither **Not found** nor **Error** connected	Connect **Error** (and **Not found** if the lookup can find nothing).
A not-found answer continued on **Error**	**Not found** is not connected	Connect **Not found**.

Message	Cause	What to do
Publishing fails although the problem list is empty	The service refused a setting the editor cannot check, for example a connection deleted meanwhile or a document type that is the name of one of your templates. The service's message names the phase and the field.	See *Refused by the service* in Messages.

Every other message is in Messages.

Related

- Overview and VIES tutorial
- Connect your systems — webhooks and **Connections**
- Conditional — branch on `restCall.` node id `.data`
- Cross-check — compare a published fact with the documents of the process
- KYB — the phase the VIES preset replaced
- Design and publish a workflow

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Send

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

A *Send* phase delivers the result of the whole process to your system: it posts a signed JSON call to the webhook endpoint you set. It is normally the last phase of a route.

Palette	**Nodes** › **Send**
Outputs	One output
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

- At the end of a route, to hand the extracted data and the outcome of each document to a CRM, an ERP or any system that can receive an HTTPS call.
- During setup, pointed at a **Test URL**, to see exactly what your system will receive before you connect it.

A *Send* phase may receive links from several phases — for example from both outputs of a Conditional, one of them through a Manual Review. It runs once, from the route the process actually took.

What happens in a run

1. The process result is closed

When the process reaches *Send*, every phase before it on its route has finished. *Send* then:

- marks every document that did not fail as **completed**;

- copies each document's extracted fields to the document management area as tags, and marks the process as completed there (both best effort: a failure here never stops the delivery);
- builds the **process result**, a JSON object with:

Key	Content
`runId`	The process (run)
`workflowId`, `version`	The workflow and the published version the process ran on
`source`	Where the process was started from
`process`	The process data supplied when it was started, if any
`summary`	`total`, `completed` and `failed` document counts
`documents`	One entry per document: `documentId`, `name`, `status`, `result`, `error`, `ecmDocumentId` and `downloadUrl`
`upstream`	The output of the phase that led into *Send*

`downloadUrl` is filled only for documents the process **produced** (for example a generated contract) and is valid for **24 hours**. Documents the customer uploaded carry only `ecmDocumentId`, which does not expire.

2. The webhook calls

With an **Endpoint URL (HTTPS)** set, *Send* makes **one call per document** the process was started with (a process without documents makes a single call with empty document fields). Each call is:

- **Method:** the **HTTP method** chosen in the panel — `POST`, `PUT`, `PATCH` or `DELETE`, `POST` when none was chosen.
- **Body** (JSON):

Key	Content
`workflowId`	The workflow
`workflowName`	The workflow's name
`documentId`	An identifier of the document in this process: `doc_` followed by 24 hexadecimal characters, stable for the same process and file name
`documentName`	The file name
`eventType`	Always `workflow.completed`
`processedAt`	UTC time of the call, as `2026-10-22T14:03:09Z`
`data`	The whole **process result** described above — the same in every call of the process

- **Headers:**

Header	Content
`Content-Type`	`application/json; charset=utf-8`
`X-Xtrakt-Run-Id`	The process (run)
`X-Xtrakt-Workflow-Id`	The workflow
`X-Xtrakt-Signature`	`sha256=` followed by the **lowercase hexadecimal HMAC-SHA256** of the raw request body, keyed with the **Signing secret**. Sent only when the phase has a secret.

Verifying the signature. Compute HMAC-SHA256 over the body bytes exactly as received (UTF-8), using the 64 characters of the signing secret **as text** — do not decode them from hexadecimal — and compare the lowercase hex result with what follows `sha256=`.

Before every call the service checks the endpoint again: it must be HTTPS on port **443**, contain no user name or password, resolve in DNS, and not resolve to a private or internal address. An endpoint that fails this check is not called.

3. The answer and the retries

Your system answers	Result
Any 2xx	Delivered. The next document is sent.
5xx, 408 or 429, a timeout, or a connection error	**Retried.**
Any other status (400, 401, 403, 404…)	**Final failure** — the run fails.
The endpoint is blocked by the check above	Final failure.

On a retryable failure the phase is run again at once, up to **3 more times** (4 attempts). A new attempt picks up where the last one stopped: documents this phase already delivered to the endpoint are **not sent again**, and the phase output lists them under `alreadyDelivered`. Retrying the phase after it completed sends nothing again either. When the attempts are exhausted the run fails.

A document can still reach your system twice if the call succeeded on your side but the answer never arrived (for example a timeout after you processed it), so keep accepting the same `documentId` twice. A correction round re-runs the phase as a new step and delivers again.

Every attempt for every document is recorded in **Webhook Deliveries** (**Administration › Integrations & Deliveries**) with the endpoint, the payload sent, the HTTP status, the response time, the outcome and an **Attempt** number that grows per document. From there **Resend webhook** sends the stored payload again, with the same HTTP method, as a new attempt. It is signed with the **signing secret of the published version that is live now** — a secret regenerated in a draft that was not published yet is not used, because your system has not been given it. If the live version no longer has that endpoint, the secret of the version the process ran on is used. A resent payload is the original one, so its `downloadUrl` links may have expired. See Connect your systems.

4. What it stores

The phase output is the process result plus a `delivery` entry: `channel` (`custom_webhook`), `event`, the **Integration name**, the endpoint, whether it was a test URL, and the HTTP status of each document's call.

When every document failed

If no document of the process succeeded and exactly one *Send* phase can still be reached, the process goes straight to that *Send*, which delivers the result with every document marked failed. The run then ends as **failed**. In this case documents are not marked completed and the document management area is not updated.

Without an endpoint

With **Endpoint URL (HTTPS)** empty, the service:

- e-mails the process result to **Recipient email**, when it is filled. The subject is **Subject**, or *Cogneris workflow result* when **Subject** is empty. A failed e-mail is retried like a webhook, and an e-mail that went out is not sent again by a retry;
- otherwise publishes the event `extract.processed` through the Cogneris vision service, with the run id, the workflow id and version, and `results` — the per-document results of the phases before *Send* (found even when a *Manual Review* wrapped them). Where that event goes is decided by that service.

Configuration

Select the phase to open **Phase Configuration**. Its header shows the phase type as `SEND`.

Field	What it does	Required	Format / values	Default	Validati messag
Description	The name shown on the node	No	Text	*Send*	—

Field	What it does	Required	Format / values	Default	Validati messag
Subject	The e-mail subject. Used only when the phase sends an e-mail (see *Without an endpoint*).	No	Text	Empty	—
Recipient email	Where the process result is e-mailed when **Endpoint URL (HTTPS)** is empty. *Previous step output will be sent as the email body.*	No	One e-mail address	Empty	*Enter a valid e-mail address. This step has an endpoint URL, so the e-mail would never be sent. Keep only one of them.* (in the pane *Send phase e-mail recipient must be valid e-mail address. Send phase has both a webhook endpoint*

Field	What it does	Required	Format / values	Default	Validati messag
					and an e mail recipient only the endpoint would be used. Ke one. (on publish)
Integration name	A name for the receiving system. Stored in the phase output; not sent in the call.	No	Text	Empty	—
Endpoint URL (HTTPS)	Where the calls go. Setting it is what makes the phase post to a webhook.	For webhook delivery	An absolute `https://` URL on port 443, or a **Test URL**	Empty	*Enter a valid HTTPS U* (in the panel) · *Send phase webhook endpoint must be valid HTTPS U* · *Send phase webhook endpoint must use the standard*

Field	What it does	Required	Format / values	Default	Validati messag
					HTTPS p 443. (on publish)
HTTP method	The method of every call to the endpoint, and of a resend.	No	POST, PUT, PATCH, DELETE	POST	*Send phase HTTP method is not supporte Use POS PUT, PAT or DELE* (on publi for a valu set outsi the edito
Description (optional)	A note on what the integration does. Not used by the service.	No	Text	Empty	—
Signing secret	The key of `X-Xtrakt-Signature`. Shown, read-only, to **administrators only** and only while an endpoint is set.	—	64 lowercase hexadecimal characters	Created when an administrator sets the endpoint; otherwise created when the workflow is published	—

Notes on the signing secret:

- Clearing **Endpoint URL (HTTPS)** as an administrator also deletes the secret.
- A save by a profile that cannot see the secret keeps the stored one.
- The secret travels with the published version: a new or regenerated secret is used by processes started **after the next publish**. Processes already running keep the secret of their version.

Buttons and actions

Button	What happens	Reversible?
Test endpoint	Sends one `POST` whose body is an empty JSON object, with the header `X-Xtrakt-Webhook-Probe: true` to the endpoint, after the same address checks as a real call. Any HTTP answer, even an error status, counts as reachable: *Endpoint reachable. The test sent a real POST with the header X-Xtrakt-Webhook-Probe: true — make your receiver ignore requests that carry it.* Otherwise *Endpoint not reachable: …*. Disabled while the URL is empty or invalid. The probe is always a `POST`, whatever **HTTP method** says: have your system answer it and discard it when it carries `X-Xtrakt-Webhook-Probe: true`.	Nothing to undo
Copy (signing secret)	Copies the secret; the button reads **Copied**. Administrators only.	—
Regenerate	Creates a new secret at once. The previous one stops being valid for processes started after you publish. Administrators only.	No — the old secret cannot be restored
Generate test URL	Creates a temporary Cogneris address that records every call made to it. Needs a saved workflow (*Save the workflow first to generate a test URL.*). One test URL per phase: generating again replaces the previous one.	**Revoke**

Button	What happens	Reversible?
Copy (test URL)	Copies the test URL; the button reads **Copied**.	—
Use as endpoint	Puts the test URL in **Endpoint URL (HTTPS)**.	Type your endpoint back
View requests	Opens **Received requests**: **Time**, **Method**, **Status**, **Size** of each call, refreshed every 10 seconds; select one for its **Headers** and **Payload**. **Refresh** reloads, **Back** returns to the list. A large payload is marked *truncated*.	—
Revoke	Ends the test URL. Calls to it then fail.	No — generate a new one

A test URL expires after **7 days** (*Expires in … day(s). History is kept for 7 days.*). Calls to it are recorded in-process with the same headers and signature a real call carries, and also appear in **Webhook Deliveries**. Replace it with your real endpoint before going live.

Common problems

Message	Cause	What to do
Enter a valid HTTPS URL.	The endpoint is not an absolute `https://` address	Correct the URL.
Send phase webhook endpoint must be a valid HTTPS URL.	Publishing refused: the saved endpoint is not a valid HTTPS URL	Correct the URL, save and publish again.
Endpoint not reachable: https_required / port_not_allowed / userinfo_not_allowed	The address is not HTTPS, not on port 443, or carries a	Use a plain `https://host/path` address on port 443.

Message	Cause	What to do
	user name or password	
Endpoint not reachable: dns_resolution_failed	The host name does not resolve	Check the host name.
Endpoint not reachable: blocked_private_address	The host resolves to a private or internal address	Use a publicly reachable address.
Endpoint not reachable: timeout / unreachable	No HTTP answer (connection, TLS or time-out)	Check that your system is up and accepts HTTPS from the internet.
Endpoint not reachable: test_hook_inactive	The test URL was revoked or expired	**Generate test URL** again.
Test URL operation failed. Try again.	Generating, loading or revoking the test URL failed	Try again.
Could not load requests.	**Received requests** could not be loaded	**Refresh**.
No requests received yet. Run the workflow to see deliveries here.	Nothing has called the test URL yet	Start a process on the published workflow.
Run error *HTTP …* with a 4xx status	Your system refused the call; not retried	Fix the receiving side, then **Resend webhook** from Webhook Deliveries.
Run error *Webhook endpoint blocked: …*	The endpoint failed the address checks at delivery time (for example its host now resolves to a private address)	Fix the endpoint, publish, and resend.

Message	Cause	What to do
Run error *Webhook POST timed out.* or *Webhook POST call failed: …* (the method named is the one configured) after the retries	Your system did not answer in 4 attempts	Check your system, then **Resend webhook**.
Run error *Test hook not found (HTTP 404). Generate a new test URL.* / *Test hook revoked or expired (HTTP 410). Generate a new test URL.*	The published endpoint is a test URL that no longer exists	Put your real endpoint, or a new test URL, and publish.
Your system rejects the signature	Wrong or outdated secret, a body changed before checking, or the secret decoded from hex	Copy the current secret, hash the raw body, use the secret as text.
Signing secret is not shown	Your profile is not administrator, or no endpoint is set	Ask your company administrator.

Related

- Connect your systems — signature verification and **Webhook Deliveries**
- Send Message — a similar phase the service never ran, no longer offered
- Conditional — choose which route reaches *Send*
- Design and publish a workflow

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

KYB

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Legacy. The KYB phase validated a company's VAT number against the EU VIES service. It was removed and replaced by the *VIES check (KYB)* connector, a pre-configured REST integration. This page explains what you see on a workflow that still has a KYB node and how to convert it.

Palette	Not offered any more. The replacement is in **Nodes** › **Pre-configured**, as the VIES connector.
Outputs	One output.
Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

When to use it

Do not add it: the palette no longer offers it. For a new VIES check, add the VIES connector from **Pre-configured** — see REST integration and Connect your systems.

You meet a KYB node only on a workflow created before the change. Such a workflow:

- **cannot be published** until it is converted;
- stops at the KYB node in any process that reaches it.

What happens in a run

The phase no longer checks anything. A process of a version published before the change that reaches a KYB node fails at that node, without calling VIES and without retrying, with this

error:

The 'kyb' node was removed. Convert the workflow with the kyb -> rest_call converter (POST /workflow/admin/migrations/kyb-to-rest-call, VIES preset) and start the case again.
(followed by the same text in Portuguese)

Processes of a converted workflow run the VIES connector instead. The review screen shows the VIES answer the same way for both, and old runs of the KYB node keep showing theirs.

Configuration

Selecting a KYB node opens **Phase Configuration** with **Description** and the legacy banner, and nothing else:

> **legacy — Legacy KYB node — convert it**
>
> *The KYB phase was removed and replaced by the "VIES check (KYB)" connector. Converting turns this workflow's KYB nodes into that connector with the same settings. Until then, this workflow cannot be published and its processes stop at this node.*

Field	What it does	Required	Format / values	Default	Validation messages
Description	Name shown on the node.	No	Text	*KYB*	—

The old settings of the node are not editable. The conversion carries them over to the connector:

KYB setting	Becomes, in the connector
The VAT number field	The VAT number source, shown as *VAT number source: …*
The default country	The default country, shown as *Default country: …* (PT is assumed when none was set)
The requester's country and VAT number	The requester, shown as *Requester: …*, only when both were set

Buttons and actions

The conversion acts on the **saved** workflow, converts **every** KYB node of it, and is available only to a Cogneris platform administrator.

Button	What happens	Reversible?
Check the conversion	Asks the service what the conversion would do, without changing anything (*Checking what would change…*). The summary shows the action, each node with its inferred settings and warnings, and anything that blocks the conversion.	Yes — nothing is changed
Convert	Converts the workflow (*Converting…*). Shown after a check whose result can be converted. On success: *Workflow converted to the VIES connector.*	No
Cancel	Closes the summary.	Yes

What the conversion does, as the summary says it:

Summary	Meaning
The draft will be converted in place.	The workflow was never published; the draft is converted.
Version … will be published with the connector; version … keeps serving the processes already running.	A new version is published with the connector; running processes stay on the old version.
Unpublished draft changes are kept as the draft, converted too; they are not published.	The workflow had unpublished edits.
The VIES connection will be created for your organization.	Your company had no VIES connection yet.
Unsaved changes in the editor will be replaced by the converted workflow.	Save first if you want to keep edits.

Summary	Meaning
There is nothing to convert in the saved workflow.	The saved version has no KYB node.

Warnings that do not block the conversion:

- *No default country was set; PT is assumed.*
- *Only one requester field was set, so the requester is ignored (as the KYB phase did).*
- *If VIES rejects the requester, the process now fails instead of retrying without it.*
- *The field was not seen in recent processes; check the source.*
- *The field appeared both in the form and in the process data; the form is used, as before.*

Common problems

Message	Cause	What to do
This workflow still has a legacy KYB node (…). Convert it to the VIES connector before publishing: open the node and use the conversion.	You tried to publish a workflow with a KYB node.	Convert it.
Save the workflow before converting it.	The workflow has never been saved.	Save it, then convert.
Only a platform administrator can convert this workflow. Ask your Cogneris administrator to convert it.	The conversion is reserved to Cogneris platform administrators.	Ask Cogneris; or delete the KYB node and add the VIES connector from **Pre-configured** yourself.
This workflow cannot be converted yet: followed by *The node has no VAT number field (every run of it fails today).*	The KYB node was never configured.	Delete it and add the VIES connector by hand.
The VAT number field cannot be used as a source.	The stored field name cannot be turned into a	Delete the node and add the VIES connector by

Message	Cause	What to do
	connector source.	hand.
A KYB setting is not valid for the connector (for example, the country code).	A stored setting is invalid.	Delete the node and add the VIES connector by hand.
The node's settings cannot be read.	The stored settings are corrupt.	Delete the node and add the VIES connector by hand.
The published version predates version history: publish it once from the editor, then convert.	The published version has no snapshot.	Publish once, then convert.
The conversion was refused; the workflow was left as it was.	The service refused the conversion.	Check again and read the reasons.
Could not complete the conversion. Try again in a moment.	The request failed.	Try again.

Related

- REST integration — the VIES connector is one.
- REST integration overview
- Connect your systems
- Manual Review — the VIES answer is shown to the reviewer.

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

REST integration — overview

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

The REST integration phase calls a REST API of one of your systems — or of a public service — in the middle of a workflow, and routes the process by the answer. This page explains the parts you configure before the phase itself (**Connections**), what each tab of the phase is for, the checks that can decide without calling, and walks through the VIES preset end to end.

Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

What it is and when to use it

A REST integration phase is made of two parts that live in different places:

Part	Where	Holds
Connection	**Administration › Connections**, shared by every workflow of your company	The base URL, the authentication and its secrets, default headers, the maximum timeout
Phase	The workflow editor, one per call	Which connection, the method and path, how the request is built from the process, how the answer is read, what is published

The base URL and the credentials live in the connection, never in the phase. The phase stores only a reference to the connection, so a credential is never copied into a workflow, its

versions or an export. The phase refuses headers and query parameters that look like credentials (see Request).

Use it to:

- look up a record (customer, supplier, tax number, registry entry) and branch on whether it was found and what it says;
- create or update a record in your system and continue only when it accepted it;
- keep the answer of an external consultation as a fact of the process, with a PDF certificate, so cross-checks and reviewers can use it.

To deliver the final result of a workflow to your system, the **Send** phase with a webhook is usually enough — see Connect your systems.

Connections

Open **Administration › Connections**. *Reusable base URLs and credentials for REST integrations. Secrets are write-only: once saved they are never shown again.*

Everyone who sees **Administration** can read the list. Only a company **administrator** sees **New connection**, **Edit** and **Delete**, and can open the form.

The list

Column	What it shows
Name	The connection's name, with a **Pre-configured** badge when it was created by a connector preset
Base URL	The base URL
Authentication	**None**, **API key**, **Basic (user and password)**, **Bearer token** or **OAuth2 (client credentials)**
Usage	*Not used in any workflow*, *Used in 1 workflow* or *Used in … workflows*
(actions)	**Edit** and **Delete**, for administrators

Loading connections… is shown while the list loads, *No connections yet.* when there is none. **Refresh** loads the list again.

The form

New connection opens *New connection*; **Edit** opens *Edit connection* with the stored values.

Field	What it does	Required	Format / values	Default	Valida messa
Name	Identifies the connection in the phase's **Connection** list	Yes	Up to 120 characters, unique in your company	—	*Enter a · The na can hav most 12 characte*
Base URL	The address every call of the phase starts from; the phase adds its **Path** to it	Yes	Must start with `https://`, without a user or password in it, and resolve to a public address	`https://`	*The bas must sta with htt and hav embedd credenti*
Authentication type	How the phase authenticates	Yes	**None**, **API key**, **Basic (user and password)**, **Bearer token**, **OAuth2 (client credentials)**	**None**	—
API key location	With **API key**: where the key is sent	Yes	**Header** or **Query string**	**Header**	—

Field	What it does	Required	Format / values	Default	Valida messa
Parameter name	With **API key**: the header or query parameter that carries the key, for example `X-API-Key`	Yes	Text	—	*Enter th parame name.*
Username	With **Basic**	Yes	Text, without `:`	—	*Enter th usernam*
Token URL	With **OAuth2**: where the access token is requested	Yes	`https://` address	`https://`	*The toke URL mu start wit https://.*
Client ID	With **OAuth2**	Yes	Text	—	*Enter th client ID.*
Scope (optional)	With **OAuth2**: sent as `scope` when requesting the token	No	Text	—	—
Audience (optional)	With **OAuth2**: sent as `audience` when requesting the token	No	Text	—	—

Field	What it does	Required	Format / values	Default	Valida messa
Client authentication	With **OAuth2**: how the client ID and secret are sent to the token URL	Yes	**In the request body** or **HTTP Basic header**	**In the request body**	—
API key / **Password** / **Token** / **Client secret**	The secret of the authentication type. Write-only: once saved it shows ••••• *set* with **Replace**. Changing the authentication type clears what you typed.	Yes, except with **None**	Text	—	*Enter th secret fo authenti type.*
Timeout (seconds, 1 to 120)	The longest any call through this connection may wait. A phase may ask for less, never more.	Yes	Whole number 1–120	30	*The time must be whole nu from 1 to*
Default headers	*Sent with every call. Mark a header as secret to store its value encrypted;*	No	Unique names; a secret header needs a value	—	*Every he needs a unique n and sec headers a value.*

Field	What it does	Required	Format / values	Default	Valida messa
	secret values are never shown again. Each row has **Header name**, **Value**, **Secret** and **Remove**; **Add header** adds a row. A stored secret header shows ••••• *set* with **Replace**.				

Save (reads **Saving…** while it works) stores the connection; **Cancel** closes the form. The service also refuses a name already used by another connection, a host name that cannot be resolved and an address that is not public; the form shows the service's message. *Could not save the connection.* is shown when the service could not be reached or gave no reason.

Stored secrets never follow a connection to a new destination. When you change the host of the **Base URL**, the **Authentication type**, the **API key location** or **Parameter name**, or the host of the **Token URL**, select **Replace** and enter the secret — and every secret header — again in the same save. As soon as you make such a change the form stops showing ••••• *set*: it asks for the secret again (*The destination changed, so the stored secret no longer applies: type it again.*) and for every secret header (*The destination changed, so the stored secret headers no longer apply: type them again.*). A change of path only, on the same host, keeps them.

Pre-configured connections. A connection created by a connector preset shows *Created from a pre-configured connector: the base URL and authentication type are fixed. Timeout, headers and secrets can still be changed.* Its **Base URL** and **Authentication type** are disabled; a save that changes them is refused with *The base URL and authentication type of a pre-configured connection cannot be changed.*

Test and delete

- **Test connection** (only while editing a saved connection; reads **Testing…**) sends one GET to the base URL with the connection's authentication and shows *Connection OK: HTTP … in … ms* or *Connection test failed: ….* It never shows a secret or the response body. Tests are rate limited per company. If the test cannot run at all: *The connection test could not be run.*
- **Delete** asks *Delete this connection?* with **Cancel** and **Delete**. A connection used by published workflows is not deleted: *This connection is used by published workflows and cannot be deleted. Unpublish or change these workflows first:* followed by the workflows, with **Dismiss**. Other failures: *Could not delete the connection.*

Loading failures: *Could not load the connections.* (the list) and *Could not load this connection.* (opening **Edit**).

The tabs at a glance

Open the phase on the canvas. The panel has seven tabs; the arrow keys move between them.

Tab	Answer this question	Details
Request	Which connection, method and path? Which parameters and headers? How long to wait, how often to retry, which statuses are a success?	Request
Schemas	What does the request body look like, what does the answer look like, and what does this phase receive?	Schemas and mapping
Request mapping	How is each field of the body filled from the process?	Schemas and mapping
Response mapping	Which parts of the answer go into the phase's `data`, and how are they normalised?	Schemas and mapping
Result	Which statuses mean "not found"? Does the body say something the status does not? Should some requests be decided without calling?	Result, Test and Publication

Tab	Answer this question	Details
Publication	Should the answer become a fact of the process? Should a PDF certificate be stored?	Result, Test and Publication
Test	Does it work with a real call?	Result, Test and Publication

A sensible order for a new integration: create the connection, fill **Request**, paste the schemas and an input example in **Schemas**, map the body and the answer, classify the answer in **Result**, **Test**, then decide what to publish.

Checks before the call

Checked in order on the request body after the de-para, before calling the destination. The first rule that matches decides and the destination is not called: Success produces the result data below, Not found continues without data, Permanent goes to Error. They sit at the bottom of the **Result** tab, under **Checks before the call**.

Use them to avoid calling a service with a request that cannot be valid — an empty identifier, a number with the wrong length — and to record a clear answer instead.

Field	What it does	Required	Format / values	Default	Validation messages
Field in the request body	The field the check reads. For POST, PUT and PATCH it is a path in the body built by the **Request mapping**; for GET and DELETE, the name of a path or query parameter. The fields of the	Yes	Dotted path; `[0]` for an item of a list, for example `items[0].code`	—	*Check … needs a field in the request body.*

Field	What it does	Required	Format / values	Default	Validation messages
	Request schema are suggested.				
Operator	How the field is compared. The list shows each operator as a symbol and its name: `equals`, `notequals`, `gt`, `gte`, `lt`, `lte`, `in`, `notin`, `contains`, `exists`, `isempty` — the same operators as the **Conditional**	Yes	One of the list	ø `isempty`	—
Value	What the field is compared with. Not shown for `exists` and `isempty`. For `in` and `notin`: *Separate the values with commas.*	Yes, for operators that take a value	Text; compared as text, ignoring case, except numbers with numbers	—	*Check … needs a value for it operator.*
Classify as	What a match decides	Yes	**Success (without**	**Success (without**	—

Field	What it does	Required	Format / values	Default	Validation messages
			calling), Not found (without calling), Permanent (error)	**calling)**	
Result data (JSON object, optional)	Only with Success (Not found continues without data): the answer to record instead of a response — *Used as the result instead of the response*, for example { "vies": { "status": "invalid", "reason": "malformed" } }. A text value written as {{request.*path*}} is replaced by that field of the request — by the value itself when the text is only the placeholder, by its text inside a longer text.	No	A JSON object, up to the size the service allows	—	*The result data of check … must be a JSON object.*

Checks are numbered **Check 1**, **Check 2**… in the order they are evaluated; **Move up** changes the order, **Remove rule** deletes one, **Add rule** adds one. With none: *No checks: the call is always made.* At most 20 checks are allowed (*At most … checks before the call are allowed.*).

A **Permanent (error)** check sends the process to **Error** with the kind `preflight` and is never retried. A check that decides **Success** goes through **Publication** like any answer.

Tutorial: check a VAT number with the VIES preset

The European Commission's VIES service confirms whether a VAT number (NIF/NIPC) is registered. Cogneris ships it as a **connector preset**: a REST integration with the connection and every tab already filled in, the same behaviour as the former **KYB** phase.

1. Add the preset

1. In the workflow editor, find the **Pre-configured** group, below the phases of the **Nodes** palette. Its entry is **VIES VAT check (EU)** with a **preset** badge; hovering shows its description: *Confirms a VAT number (NIF/NIPC) with the European Commission's VIES service and publishes the answer as the kyb_vies document, with a PDF certificate. Same behaviour as the KYB phase.*

2. Select it, or drag it onto the canvas. A form titled **VIES VAT check (EU)** opens. The phase is created only when this form succeeds.

3. Fill in the parameters (required ones are marked with `*`):

Parameter	What it is	Required	Default
VAT number source	Where the VAT number is read from. Pick a source from the list or type one: `$form.` + the form answer key, `$process.` + a process key, or `$doc.` + document type + `.` + field.	Yes	`$form.nif`
Default country	The country assumed when the number has no two-letter prefix. *Two-letter country code, for example PT.* Typed letters become capitals.	Yes	`PT`

Parameter	What it is	Required	Default
Requester country	Your own country, when VIES should record who asked	No — but required when **Requester VAT number** is set	—
Requester VAT number	Your own VAT number, for the same purpose	No — but required when **Requester country** is set	—

4. Select **Add to the flow** (it reads **Preparing the connector…** while it works). **Cancel**, or Escape, closes the form without adding anything.

Messages of the form: *Fill in this parameter.*, *Use a two-letter country code in capitals, for example PT.*, *This source is not valid: use $form…., $process…. or $doc……..*, and, from the service, *The value of "…" was refused. Check it and try again.*, *One of the parameters was refused. Check them and try again.*, *This preset is no longer available. Reload the page to update the list.*, *Could not prepare the connector. Try again in a moment.*

The connection. The first time the preset is added in your company, Cogneris creates the connection **VIES (European Commission)** — base URL `https://ec.europa.eu/taxation_customs/vies`, authentication **None**, timeout 20 seconds, marked **Pre-configured**. Only an administrator can cause that creation; anyone else sees *This connector needs a connection that does not exist yet in your organization, and only an administrator can create it. Ask an administrator to add this preset once; after that you can use it.* After that, everyone who edits workflows can add the preset, and every VIES phase reuses the same connection. A company that added the preset before version 2 keeps its connection as it is, under its earlier name **VIES (Comissão Europeia)**: the preset finds it by its key, not by its name. Phases added before version 2 keep their settings until the preset is added again.

2. What the preset fills in

The new phase is a normal REST integration with a **preset** badge on the canvas and, at the top of its panel, the banner *Pre-configured connector: VIES VAT check (EU) (version 2)* and the button **Preset parameters**.

Tab	Filled with
Request	**Connection** VIES (European Commission); **Method** POST; **Path** `/rest-api/check-vat-number`; no path or query parameters, no headers; **Timeout (seconds)** 20; **Retries** 5; **Wait between retries (seconds)** 600; Idempotency-Key off; **Success status codes** 200; **Not found status codes** 404
Request mapping	`countryCode`: the first two letters of the number when it starts with them, in capitals, otherwise the **Default country**. `vatNumber`: the number without spaces, punctuation or that prefix. With a requester: `requesterMemberStateCode` and `requesterNumber` as fixed values. A helper field `malformed`, written only when the number cannot be valid: with **Default country** PT, a Portuguese number — one with the PT prefix or with no prefix — that is not 9 digits; with another default country, a PT-prefixed number that is not 9 digits.
Result — checks before the call	Rule 1: `vatNumber` `isempty` → **Permanent (error)**: a process without a number goes to **Error** without calling VIES. Rule 2: `malformed` `exists` → **Success (without calling)**, with the result `vies.status` *invalid* and `vies.reason` *malformed*, echoing the country and number.
Result — rules on the body	**Not-found status codes (4xx)** empty: no status means not found. `errorWrappers[0].error` in `INVALID_INPUT` → Success (the number is invalid). `INVALID_REQUESTER_INFO` → Permanent when a requester is configured, Success otherwise. `actionSucceed` equals `false` → Transient. `valid` isempty → Transient.
Response mapping	`vies.status` (`valid`, `invalid` or `unknown`), `vies.countryCode` and `vies.vatNumber` (as sent), `vies.name` and `vies.address` (empty or `---` become null), `vies.nameDisclosed`, `vies.requestDate`, `vies.requestIdentifier`, `vies.requesterAccepted`, `vies.reason` (`not_registered` when VIES says the number is not valid, or the VIES error code)
Publication	**Publish the result** on, **Document type** `kyb_vies`, **Display name** *VIES check*, published on Success only. **Generate the PDF proof** on,

Tab	Filled with
	Layout VIES (VAT check), **File name** `kyb-vies-` followed by the country and the number.

`kyb_vies` is the document type the former KYB phase produced, so cross-checks, templates and the review screens that read KYB results read this one too. It is the one published type allowed to have the same name as an existing document type.

3. What you still have to do

1. **Check the source.** Make sure **VAT number source** names the field where your process really has the number. To change it, select **Preset parameters**, change the value and select **Apply parameters**. *Applying the parameters again replaces this node's configuration with the preset's. Its connections to other nodes are kept.* — any manual change made in the tabs is lost.
2. **Connect the outputs.** Connect **Success** to the next phase. Connect **Error**: without it, a process with no number, a refused requester or VIES unavailable after the five retries fails. An HTTP 404 from VIES produces **Not found**: connect that output if you want such a process to continue on its own path; left unconnected, the process stops there.
3. **Use the answer.** Branch with a **Conditional** on `restCall.` node id `.data.vies.status`, or compare the `kyb_vies` fact with the documents in a **Cross-check**. The certificate is stored with the process documents.
4. **Optionally test.** In **Schemas**, give an **Input example** that carries the source, for example { "$form": { "nif": "PT500000000" } }, then use **Test** › **Test the call** › **Call now**. The call really reaches VIES.
5. **Save and publish** the workflow.

Expect long waits when VIES is unavailable: a transient answer is retried five times, ten minutes apart.

If the preset catalog cannot be loaded later, the banner shows *The preset catalog is not available right now, so its parameters cannot be reopened. The node keeps working as configured.* and **Preset parameters** is disabled. A phase created from a workflow template whose preset parameters were never applied shows *VIES VAT check (EU): fill in the parameters to finish this connector.* — select **Preset parameters** and apply them.

Requester refused by VIES

The former KYB phase retried without the requester when VIES refused it. The preset does not: with a requester configured, `INVALID_REQUESTER_INFO` goes to **Error**, and its code is in `restCall.` node id `.error.bodyExcerpt`. To keep the old behaviour, connect **Error** to a **Conditional** that checks that field contains `INVALID_REQUESTER_INFO` and, when true, to a second VIES phase added without requester parameters.

Related

- REST integration — the phase page
- Request · Schemas and mapping · Result, Test and Publication · Messages
- KYB — converting the former phase to this preset
- Connect your systems

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

REST integration — Request

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

The **Request** tab of the REST integration says where the call goes and how it is sent: the connection, the method, the path and its parameters, query parameters, headers, the timeout, the retries and which statuses count as a success. The request **body** is built in the **Request mapping** tab — see Schemas and mapping.

Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

Fields

Field	What it does	Required	Format / values	Default	Valid mess
Connection	The connection the call goes through: base URL, authentication, default headers and maximum timeout. Once chosen, *Base URL …* · *authentication:* … shows the connection's	Yes	One of your company's connections	**Choose a connection** (none)	*Choose connec Could r load th connec* (the list not loa

Field	What it does	Required	Format / values	Default	Valid mess
	address and authentication type. *The base URL and the credentials live in the connection, never in the phase.* **Manage connections** opens **Connections**.				
Method	The HTTP method	Yes	GET, POST, PUT, PATCH, DELETE. GET and DELETE send no body, so the request schema and the **Request mapping** do not apply to them.	POST	—
Path	Added to the connection's base URL. *Relative to the connection base URL. Each name in curly braces*	Yes	Starts with `/`; no scheme, host, `?` query, `#` fragment, `@`, spaces, `\`, `.` or	—	*Fill in th path. · T path m relative connec no sche host. · T path m*

Field	What it does	Required	Format / values	Default	Valid mess
	becomes a path parameter. For example `/customers/` followed by a name in curly braces creates a path parameter with that name.		`..` segments, or encoded slashes; up to 2048 characters. A parameter name starts with a letter or `_` and has letters, digits, `_` or `-`.		*start w and ca contain query (fragme an at si spaces backsla "." or ".." segme encode slashes path parame must b name i braces starts v letter o unders and use letters, unders and hy*
Path parameters	One entry per name in curly braces in **Path**, in order. Entries appear and disappear as you edit the path; they cannot be renamed or removed here.	One per name in the path	A source, or a first transform that produces a value (`constant` , `concat` , `coalesce`)	*The path has no parameters.*	*The pa parame has no The pa parame needs a source.*

Field	What it does	Required	Format / values	Default	Valid mess
	Each entry has a **Source** and **Transforms** (see below). The value is URL-encoded before it is placed in the path.				
Query parameters	Added to the URL as `?name=value&…`, in order. Each has **Name**, **Fixed value** (a check box), then either **Value** (when fixed) or **Source** and **Transforms**. **Add query parameter** adds one; **Remove** deletes it. An unnamed parameter is titled *New parameter*. A parameter whose value is empty at run time is left out of the URL.	No	Up to 50. **Value** takes text, a number, `true`, `false` or `null`; wrap it in quotes to force text.	*No query parameters.*	*Every q parame needs a name. · query parame needs a source fixed va The que parame looks lik credem cannot in the p and is r saved. connec type AP in the q*

Field	What it does	Required	Format / values	Default	Valid mess
Headers	Extra headers sent with this call. *Non-secret headers only. Authentication comes from the connection and cannot be set here.* Each has **Header name**, **Header value** and **Remove**; **Add header** adds one.	No	Up to 50; values up to 4096 characters, on one line	—	*Every h needs a name. · header cannot in the p authent and sec belong connec*
Timeout (seconds)	How long one attempt waits for an answer. The connection's timeout is the ceiling: the shorter of the two applies.	Yes	Whole number 1–120, not more than the connection's timeout	30	*The tim must be betwee and 120 second timeou cannot longer t the connec (... seco*
Retries	How many times a transient failure is tried again after the first attempt. 0 means no retry.	Yes	Whole number 0–10	3	*Retries be betw and 10.*

Field	What it does	Required	Format / values	Default	Valid mess
Wait between retries (seconds)	The wait before each retry. When the destination answers with a `Retry-After` longer than this, that wait is used instead, up to 600 seconds.	Yes	Whole number 1–600	10	*The wa betwee retries be betv and 60 second*
Send an Idempotency-Key that stays the same across retries	Sends an `Idempotency-Key` header, the same on every attempt of the same process and phase, so a destination that supports it does not apply the request twice. A correction round of the process sends a new key.	—	On or off	On	—
Success status codes	The statuses that mean the call succeeded. Any other status is classified as described	Yes	Codes separated by commas or spaces, each between	`200, 201, 202, 204`	*Enter a one suc status … is not succes status succes*

Field	What it does	Required	Format / values	Default	Valid mess
	under *What happens in a run* in REST integration.		200 and 299		*codes r be betv 200 an*

Below the retry fields, *Transient failures (5xx, 408, 429, timeout) are retried up to … times, every … seconds or as the server asks.* repeats your settings. Below **Success status codes**: *Success continues the flow; Not found receives lookups that found nothing (see the Result tab); Error receives permanent failures and exhausted retries. An unconnected Not found falls back to Error; with Error unconnected too, the run fails.*

Source and transforms of a parameter

A path or query parameter is filled like a field of the request body, without a target:

- **Source** — *Search a source path*. Typing filters the list (*No matching paths* when nothing matches); arrow keys move, Enter picks, Escape closes. You can also type a path that is not in the list. The list offers the paths of the **Input example** and the process data: `$form.` + a form answer key, `$process.` + a process key (`$process.today` always), `$doc.` + document type + `.` + field (and the `[]` form for every document of the type), `$run.id`, `$run.workflowId`, `$run.tenantId`. See *Sources* in Schemas and mapping.
- **Transforms** — applied in order to the value, for example `digits_only` to strip punctuation from an identifier. *No transforms: the value is copied as is.* The transforms are described in Schemas and mapping.

A path parameter whose value is missing at run time stops the request before it is sent: the process goes to **Error** with the kind `request_schema`. A path parameter whose value is empty, `.` or `..` is refused the same way. A transform that fails on a path or query parameter has the same effect. A query parameter whose value is missing is simply left out.

Names you cannot use

These names are refused because a credential typed here would be stored in clear in the workflow:

Where	Refused names (any case)
Headers	`Authorization`, `Proxy-Authorization`, `Cookie`, `X-API-Key`, `Api-Key`, `Idempotency-Key`, and the header that carries the connection's API key
Query parameters	`api_key`, `apikey`, `api-key`, `access_token`, `token`, `key`, `password`, `secret`, `client_secret`, `signature`, `sig`

A refused header or query parameter stays visible with its message while you fix it, but it is **never saved** with the workflow. Put the credential in the connection instead: an **API key** connection with **API key location** set to **Query string** sends the key as a query parameter.

How the request is sent

- **URL** — the connection's base URL, the **Path** with its parameters filled in, then the query parameters; an **API key** connection in the query string adds its key last. A request that would leave the connection's host or base path is refused (kind `ssrf`).
- **Headers**, in this order, each replacing an earlier one with the same name: the connection's default headers, the connection's secret headers, the phase's **Headers**, the authentication of the connection, then `Idempotency-Key`. Every request asks for `application/json`.
- **Body** — for POST, PUT and PATCH, the JSON object built by the **Request mapping**, sent as `application/json; charset=utf-8`. Without mapping rules the body is an empty JSON object.
- **Credentials** are decrypted only at the moment of the call and removed from everything the phase keeps: its output, its error excerpt, the test view and the logs.

Common problems

Message	Cause	What to do
Could not load the connections.	The list of connections did not load	Reload the editor; if it persists, contact support.
The connection you need is not in **Connection**	It does not exist yet, or was deleted	Ask an administrator to create it in **Connections**.

Message	Cause	What to do
The path must be relative to the connection: no scheme or host.	**Path** starts with a scheme such as `https:` or with `//`	Put the scheme and host in the connection's **Base URL**; keep only the path here.
The path must start with "/"… in the problem list	The path has no leading `/`, or carries a query, a fragment or another character the service refuses	Start **Path** with `/` and move query values to **Query parameters**.
… is not a success status code…	A success code such as 302 or 404	Keep only 2xx codes in **Success status codes**; use **Not-found status codes (4xx)** in **Result** for a not-found status.
The timeout cannot be longer than the connection's…	**Timeout (seconds)** is larger than the connection's timeout	Lower it, or raise the connection's **Timeout (seconds, 1 to 120)**.
The process waits a long time at the phase	Transient failures are being retried	Check `restCall.` node id `.error` in the process; lower **Retries** or **Wait between retries (seconds)** if the wait is too long.

Every message of the REST integration is listed in Messages.

Related

- REST integration
- Overview and VIES tutorial — **Connections**
- Schemas and mapping — the body, sources and transforms
- Result, Test and Publication

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

REST integration — Schemas and mapping

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Three tabs of the REST integration describe the data: **Schemas** says what the request body and the answer look like and what the phase receives; **Request mapping** builds the body from the process; **Response mapping** translates the answer into the phase's `data`. Both mappings (the *de-para*) are lists of rules: each rule reads a **source**, applies **transforms** in order and writes a **target**.

Content status	From source; not observed (`workflow-fe` `f88c5c1`, `workflow-be` `6703373`, 2026-10-22)

Schemas

Field	What it does	Required	Format / values	Default	Validation messages
Request schema	Describes the request body. Its fields become the targets of the **Request mapping**, and before every call the body is validated against it. Shown only for POST, PUT and PATCH; for GET and DELETE: *… requests have no body, so there is no request schema or request mapping.*	No	A JSON Schema (draft-07 or 2020-12) describing an object or an array	—	*Invalid JSON: … · Not a valid schema: describe an object (type "object") or an array (type "array").*
Response schema	Describes the answer. Its fields are offered as sources of the **Response mapping** and as fields of the **Result** rules, and a success answer is validated against it.	No	Same as above	—	Same as above
Input example (JSON)	*An example of what this phase receives from the previous one. It feeds the source paths of the mapping and the test; it is never*	No	JSON	—	*Invalid JSON: …*

Field	What it does	Required	Format / values	Default	Validation messages
	used at run time. It may also carry the process data under `$form`, `$process`, `$doc` and `$run`, for example { "$form": { "nif": "PT500000000" } }; their keys are then offered as sources and the **Test** uses their values.				

Schema input

Each schema has two modes, chosen under **Schema**:

- **JSON Schema** — *Paste the JSON Schema*. The schema is applied as soon as the text is valid; clearing the text removes the schema.
- **Example → infer** — *Paste an example JSON*, then select **Infer schema**. The editor writes a schema from the example, switches back to **JSON Schema** and shows *Schema inferred from the example. Review it before saving.*

How a schema is inferred from an example:

In the example	In the schema
Text	`string`
A whole number, a decimal number	`integer`, `number` (both seen in the same place: `number`)
`true` / `false`, `null`	`boolean`, `null`

In the example	In the schema
An object	`object` with one property per key
A list	`array`; the items of the list are merged into one item schema, so an item that has a key another lacks still contributes it
Different types in the same place	A list of types, for example `string` and `null`

An example cannot prove that a field is mandatory, so an inferred schema marks nothing as required. Add `required` by hand where a field must always be present: required fields of the **Request schema** must then have a rule.

The service limits a schema to 256 KB, 64 levels and 20000 nodes, and does not support `patternProperties`.

Load an example from a run

Load from a run (reads **Loading runs…**) lists up to five recent outputs of the phase that feeds this one, as **Use run example … …** with the number and the time. Picking one fills **Input example (JSON)** and shows *This example came from a real run and is used only while editing: it is not saved in the workflow.* The process data of those runs offers its keys as sources; its values are masked by the service.

Message	Cause
Save the workflow before loading an example from a run.	The workflow was never saved; the button is disabled
No recent run reached this phase yet.	No process has completed the previous phase yet
Loading from a run is not available in this environment yet.	The service does not offer it here
Could not load the recent runs.	Any other failure; try again

Request mapping

Builds the request body: the target is the request schema, the source is the input example. Only for POST, PUT and PATCH.

With a request schema, the tab shows the schema as a tree (**Destination fields**): each field with its name, its type, **Required** when the schema requires it, and **No rule** when nothing writes it yet. Fields that hold a value, and lists (shown with `[]`), can be mapped:

Button	What happens	Reversible?
Map	Creates a rule for the field and opens it	Yes — **Remove**
Edit / **Close**	Opens or closes the rule's form	—
Remove	Deletes the rule	No

A mapped field shows *Source: … · transforms: …* (*(no source)* when it has none). A rule whose target is not in the schema is listed under **Rules without a matching schema field**, with an editable **Target path**.

Without a request schema, rules are a plain list: **Add rule** adds one with an empty **Target path**, and each has **Edit** and **Remove**.

At the top, *No issues found.* or *Issues to fix: …* counts the errors of all the rules. Each rule's form shows its own messages; they are listed in Messages.

The rule form

Field	What it does	Required	Format / values	Default	Validat messag
Target path	Where the value is written in the body (or in `data` for a response rule). Set by the	Yes	Dotted path: `customer.name`; `items[0].code` for a fixed position; `items[]` (last segment only) for a list rule	—	*The rule has no target p · Target does no exist in t schema.*

Field	What it does	Required	Format / values	Default	Validat messag
	schema tree, or typed for a rule without a schema field.				
Source	Where the value is read. See Sources.	Yes, unless the first transform is `constant`, `concat` or `coalesce`	A path, or a reserved source	—	*A source required unless t first transfor constan concat coalesc Source ' was not found in the sam* (a warni the rule still wor
Transforms	Applied in order to the value read from the source — see Transforms	No	**Add transform** › **Choose a type**; per transform: **Type**, **Move up**, **Move down**, **Remove transform**	*No transforms: the value is copied as is.*	Per transfor below
Item rules for …	For a list target (ending in `[]`):	No	Rules, as above; sources inside are relative to the item	—	*Item rule only app to array targets*

Field	What it does	Required	Format / values	Default	Validat messag
	rules applied to each item of the source list, with paths relative to the item. `$` alone is the item itself. Lists can be nested up to 3 levels.				*(ending []). · Arra can be nested u to 3 leve*

Every required field needs a rule. A field the schema requires, inside an object that is required or that you map, must be the target of a rule: *Required field "…" has no rule.*

How rules are applied

- Rules run in order. Each reads its source, applies its transforms and writes the target, creating the objects on the way.
- When the value is **missing** at the end — the source does not exist, a `regex_extract` did not match, a `concat` found nothing — the target is **not written**. Add a `default` transform to write something instead.
- A value of `null` produced on purpose (by `constant`, `default`, `lookup` or the source itself) **is** written, as `null`.
- Two rules may write the same target: the later one replaces the earlier one when it produces a value. The VIES preset uses this to fill `vies.status` from two different fields of the answer.
- A **list rule** needs a source and a target that both end in `[]`, and takes no transforms: without item rules it copies the list; with item rules it builds one new item per source item.
- At most 500 rules per mapping, paths of up to 512 characters and 32 segments, fixed positions up to 10000.

Response mapping

Translates the response into the phase output under data: the source is the response schema, the target is a free path. Without rules, data receives the raw response.

There is no target schema on this side: rules are a plain list with **Target path**, **Add rule**, **Edit** and **Remove**. The sources offered are the fields of the **Response schema**. Besides paths in the answer, a response rule may read **what was sent**, with `$request.` followed by a path: the body built by the **Request mapping**, or, for GET and DELETE, a path or query parameter by name. The VIES preset uses `$request.countryCode` to return the country it asked about. No other reserved source is allowed here: *Source "…" is not valid here: a response rule can only read what was sent, with $request.….*

Order of work on a success answer: the **Rules on the response body** of the **Result** tab first, then the **Response schema**, then the response mapping.

Sources

Source	Reads	Where it is allowed
A path such as `customer.taxId`, `items[0].code`, `items[]`	The input of the phase — the output of the previous phase — as in the **Input example**. Inside item rules, a path relative to the current item.	Everywhere
`$form.` + key	An answer of the portal form, with dotted keys for nested answers	Top-level request rules, path and query parameters
`$process.` + key	Process metadata, for example `$process.identifier`, `$process.requester.name`, `$process.today`	Same
`$doc.` + document type + `.` + field	That field of the first document of that type that has it	Same

Source	Reads	Where it is allowed
`$doc.` + document type + `[].` + field	The list of that field over every document of that type	Same
`$run.id`, `$run.workflowId`, `$run.tenantId`	Run metadata, with no personal data	Same
`$request.` + path	What the phase sent	Response rules only

Values read from `$form`, `$process`, `$doc` and `$run` are text. A key that is absent resolves to nothing, so a `default` transform applies. Document types are your extraction templates' names; the list offers their fields. They are refused inside item rules — *Source "…" reads the run ($form, $process, $doc, $run) and can only be used in top-level rules, not in item rules.* — and a malformed one is refused everywhere: *Source "…" is not valid: use $form…., $process…., $doc……., $doc….[]…. or $run.id, $run.workflowId, $run.tenantId.*

Transforms

The transform list is closed: the **Type** list shows the sixteen types by name — for example *Extract with regex* for `regex_extract`, *First non-empty value* for `coalesce`, *Look up in table* for `lookup`. The headings below use the technical name the configuration stores. Transforms run in order, each on the result of the one before.

Most transforms leave a missing or `null` value as it is — only `constant`, `default`, `concat` and `coalesce` produce a value from nothing. Put `default` last to fill what is still missing.

A transform that **fails** at run time stops the mapping: the process goes to **Error** with the kind `request_schema` (request side, path and query parameters) or `response_schema` (response side). The message, in English, names the rule, the transform and the value, cut at 100 characters — for example *Rule 'customer.document': to_number failed for the value 'abc'.*

`constant` — fixed value

Replaces the value with a fixed one, whatever was read. As the first transform, the rule needs no source.

Parameter	Format
Value	*Text, number, true, false or null. Wrap in quotes to force text.* `12` is the number 12, `"12"` the text 12, `abc` the text abc.

Example: any input → **Value** `"BR"` → `"BR"`. With **Value** `null` the target is written as `null`.

`default` — value when empty

Uses **Value** (same format as `constant`) when the value is missing, `null` or empty text; otherwise keeps it.

Example: missing → **Value** `0` → `0`; `"abc"` → `"abc"`.

`trim`, `upper`, `lower`

Remove the spaces at both ends, or change to capitals or small letters. They act on text only; any other value is kept as it is. No parameters.

Example: `" pt "` → `trim` → `"pt"` → `upper` → `"PT"`.

`digits_only`

Keeps only the digits 0–9 of a text or a number; the result is text. No parameters. Fails on an object, a list or a true/false value.

Example: `"12.345.678/0001-90"` → `"12345678000190"`.

`alnum_only`

Keeps only letters (of any alphabet) and digits of a text. Other values are kept as they are. No parameters.

Example: `"PT 500-000-000"` → `"PT500000000"`.

`regex_extract` — regular expression

Returns one group of the first match of a regular expression. *Returns the chosen group. With no match the value is empty, so a following default can fill it.* A number is matched as its text.

Parameter	Format	Default
Regular expression (up to 256 characters)	A .NET regular expression	—
Capture group (0 = whole match)	A whole number from 0 to the number of groups in the expression	1
Ignore upper/lower case	On or off	Off

Example: `"PT500000000"`, expression `^([A-Z][A-Z])`, group 1 → `"PT"`. With no match → missing.

Errors: *regex_extract needs a pattern of 1 to 256 characters.* · *regex_extract: the group must be a whole number from 0 up to the number of groups in the pattern.* · an expression the service cannot compile is refused at publication · a match that takes longer than 100 ms fails at run time.

`concat` — concatenation

Joins the values of several source paths with a separator. It ignores the value read by the rule, so as the first transform the rule needs no source. Missing and `null` values are skipped; numbers and true/false are joined as text; an object or a list fails. When every source is missing, the result is missing.

Parameter	Format	Default
Source paths (one per line)	Paths, as in **Source**; at the top level, reserved sources are allowed	—
Separator	Text, may be empty	empty

Example: `firstName` = `"Ana"`, `lastName` = `"Silva"`, separator a space → `"Ana Silva"`.

Error: *concat needs at least one source path.*

`coalesce` — first non-empty

Uses the first path whose value is neither empty nor null. Like `concat`, it ignores the value read by the rule. When every path is missing, `null` or empty text, the result is missing.

Parameter	Format
Source paths (one per line)	Paths, as in **Source**

Example: `mobile` empty, `phone` = `"+351 210 000 000"` → `"+351 210 000 000"`.

Error: *coalesce needs at least one source path.*

`date_format` — date with format and time zone

Reads a date in one format and writes it in another.

Parameter	Format	Default
Input format (iso or pattern)	`iso` for ISO 8601, or a .NET date pattern such as `dd/MM/yyyy`	`iso`
Output format (iso or pattern)	`iso` (written as `yyyy-MM-ddTHH:mm:ssK`) or a pattern	`iso`
Time zone (optional)	A time zone name such as `Europe/Lisbon`	none

With a time zone, a date that carries no offset is read as a time in that zone, and a date that carries one is converted to it.

Examples: `"15/03/2024"`, input `dd/MM/yyyy`, output `yyyy-MM-dd` → `"2024-03-15"`. `"2024-03-15T13:45:30Z"`, input `iso`, output `dd/MM/yyyy HH:mm`, time zone `Europe/Lisbon` → `"15/03/2024 13:45"`.

Errors: *date_format needs both the input and output formats.* · a pattern or time zone the service does not recognise is refused at publication · a value that is not text, or that does not match the input format, fails at run time.

`number_format` — number with separator and decimals

Reads a number written with a given decimal separator, with or without thousands separators, and returns a JSON number.

Parameter	Format	Default
Decimal separator of the input	`.` or `,`. The other sign is taken as the thousands separator.	`,`
Decimal places (optional)	Whole number 0–10: rounds, halves away from zero	none (no rounding)

Examples: `"1.234,56"` with `,` → `1234.56`; with 1 decimal place → `1234.6`. `"1,234.5"` with `.` → `1234.5`. A number is kept (and rounded if asked).

Errors: *The decimal separator must be "." or ",".* · a text that is not a number in that format fails at run time.

`to_string`, `to_number`

- `to_string` writes a number or true/false as text; text is kept; an object or a list fails. Example: `123` → `"123"`.
- `to_number` reads text as a number with `.` as the decimal separator; a number is kept; anything else fails. Example: `"42.5"` → `42.5`. For `"42,5"` use `number_format`.

No parameters.

`to_boolean` — boolean with true and false values

Turns text or a number into `true` or `false`, comparing without regard to case. A value already `true` / `false` is kept.

Parameter	Format	Default in the editor
True values (comma-separated)	Texts	`true, 1, sim, s, yes, y`
False values (comma-separated)	Texts	`false, 0, nao, não, n, no`

Example: true values `sim, yes, 1`, false values `não, no, 0`: `"Sim"` → `true`, `0` → `false`, `"talvez"` → fails.

Errors: *to_boolean needs at least one true value and one false value.* · the same text in both lists is refused at publication · a value in neither list fails at run time.

`lookup` — value table with fallback

Replaces a value by the one a table gives for it. The comparison is exact (case matters); a number or true/false is looked up as its text.

Parameter	Format	Default
Value table	Pairs **From value** → **To value**; **Add pair**, **Remove pair**. **To value** is text.	—
When the value is not in the table	**Keep the original value**, **Use null**, **Raise an error**, **Use a fixed value**	**Keep the original value**
Fixed value	With **Use a fixed value**: the value to use, in the format of `constant`	—

Example: table `M` → `Male`, `F` → `Female`, fallback **Use a fixed value** `"Unknown"`: `"F"` → `"Female"`, `"X"` → `"Unknown"`. With **Raise an error**, `"X"` fails.

Errors: *lookup needs at least one entry in the table.* · *The lookup fallback is not valid.* · a value not in the table with **Raise an error**, or an object or a list, fails at run time.

Other messages

Unknown transform "…". appears when a stored configuration names a type that does not exist; the transform is shown but cannot be edited. Remove it.

Related

- REST integration
- Request — path and query parameters use the same sources and transforms
- Result, Test and Publication
- Messages

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

REST integration — Result, Test and Publication

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

The last three tabs of the REST integration: **Result** decides which output an answer takes, **Test** makes one real call with the current settings, and **Publication** decides what the process keeps of the answer.

Content status	From source; not observed (`workflow-fe` `f88c5c1` , `workflow-be` `6703373` , 2026-10-22)

Result

The tab opens with: *This phase has three outputs: Success, Not found and Error. Flows saved with only Success and Error keep working: while Not found is unconnected, a not-found result goes to Error, and never to Success.*

Field	What it does	Required	Format / values	Default	Validation messages
Not-found status codes (4xx)	*A response with one of these statuses goes to the Not found output. Leave the field empty to treat them as errors.*	No	Codes between 400 and 499, separated by commas or spaces	`404`	*… is not a 4xx status code: not-found codes must be between 400 and 499.*
Rules on the response body	Rules that read the body of a response whose status is in **Success status codes**. *Checked in order on the body of a success response, before the response schema and the de-para; the first rule that matches decides. Transient is retried like a server error; Permanent goes to Error.* With none: *No rules: the status code alone decides the output.*	No	Up to 20 rules	none	*At most … result rules are allowed.*

Each rule, titled **Rule 1**, **Rule 2**…:

Field	What it does	Required	Format / values	Default	V n
Field in the response	The field of the body the rule reads. The fields of the **Response schema** are suggested.	Yes	Dotted path; `[0]` for an item of a list, for example `errorWrappers[0].error`	—	*F r f r*

Field	What it does	Required	Format / values	Default	V n
Operator	How the field is compared. The list shows each operator as a symbol and its name in your language, for example = *equals*, ≠ *does not equal*, ∈ *is one of*. They are the operators of the **Conditional**: `equals`, `notequals`, `gt`, `gte`, `lt`, `lte`, `in`, `notin`, `contains`, `exists`, `isempty`	Yes	One of the list	*? exists*	-
Value	What the field is compared with. Not shown for `exists` and `isempty`.	For the operators that take one	Text. Compared as text, ignoring case; `gt`, `gte`, `lt`, `lte` compare numbers.	—	*F r v c*

Field	What it does	Required	Format / values	Default	V n
	For `in` and `notin`: *Separate the values with commas.*				
Classify as	What a match means	Yes	**Success**, **Not found**, **Transient (retry)**, **Permanent (error)**	**Transient (retry)**	-

Move up changes the order, **Remove rule** deletes a rule, **Add rule** adds one at the end.

How the operators treat a field that is not in the body: `exists`, `equals`, the comparisons, `contains`, `in` and `notin` are false; `isempty` and `notequals` are true. `isempty` is also true for `null`, empty text, an empty list and an empty object.

When to use rules: for services that answer 200 with an error inside the body, or with an empty result that means "nothing found". For example, a rule `results` `isempty` → **Not found**, or `status` `equals` `BUSY` → **Transient (retry)**.

A success-status response with an empty body is a Success with no data. A body that is not JSON is an error of kind `response_schema`.

Checks before the call

The same tab ends with **Checks before the call**: rules on the request that decide without calling the destination. They are described, with every field, in Overview and VIES tutorial.

Test

Sends one real request with the current settings and the input example, without retries.

Button	What happens	Reversible?
Test the call	Asks for confirmation: **Confirm the real call** — *This call will really be made to the target system.*	—

Button	What happens	Reversible?
	Data may be created or changed there. Continue? Reads **Calling…** while the call runs.	
Call now	Makes the call	**No** — the destination receives it and may create or change data
Cancel	Closes the confirmation without calling	—

Test the call is disabled — *Choose a connection and a valid path before testing.* — while there is no connection, the path is empty or absolute, or a header or query parameter has a refused name. Without an **Input example**: *There is no input example: mappings will read an empty input.*

The test uses the settings on screen, saved or not, and the **Input example** as the phase's input — including any `$form`, `$process`, `$doc` or `$run` it carries. It goes through the same steps as a run — checks before the call, the request schema, the call, the rules on the body, the response schema and the response mapping — but makes **one** attempt, publishes no fact and generates no certificate.

The result shows three columns:

Column	Content
Sent	The method and the URL (an API key sent in the query shows as `***`), and the body. When a check before the call decided, the URL is empty: nothing was sent.
Response	*Status …* and the body, with the connection's secrets removed. `–` when nothing came back.
Phase output	What the phase would pass on, including `restCall.` node id with `ok`, `found`, `status`, `data` or `error` — see REST integration

Message	Cause	What to do
Too many tests in a short time. Wait a minute and try again.	More than 10 tests in a minute for your company	Wait a minute.
The test call is not available in this environment yet.	The service does not offer the test here	Test in a process instead.
The test could not be run. Check the settings and try again.	The service refused the settings — every check made at publication applies — or the call could not be made	Check the problem list and the settings the editor does not check: see *Refused by the service* in Messages.

A test that reached the destination but failed is **not** a message: it shows in **Phase output** as `error`, with its kind and message.

Publication

Publish as a process fact

Saves the mapped result as a document of the run, so cross-checks and later steps can use it by its document type.

Field	What it does	Required	Format / values	Default	Validation messages
Publish the result	Turns publication on or off. Off, the answer stays only in the phase's output.	—	On or off	Off	—
Document type	The type the fact is published	Yes, when publishing	`a` – `z`, `0` – `9`, `_`; starts	—	*The document type must start with a*

Field	What it does	Required	Format / values	Default	Validation messages
	under. A **Cross-check** role or a document template addresses the fact by this type. *Lowercase letters, digits and underscores, starting with a letter (2 to 64 characters), for example kyb_vies. It must not be the type of one of your templates.*		with a letter; 2–64 characters		*lowercase letter and use only lowercase letters, digits and underscores (2 to 64 characters).* · a type equal to the name of one of your templates (or to its lowercase, accent-free form) is refused at publication
Display name (optional)	The name of the fact among the documents of the process. Empty, the document type is shown.	No	Up to 120 characters, one line	—	*The display name can have at most … characters.*
Publish when the result is	The outcomes that publish: **Success**, **Not found**, **Error**. *By default*	At least one	Check boxes	**Success**	*Choose at least one result to publish.*

Field	What it does	Required	Format / values	Default	Validation messages
	only a successful result is published.				

What is published is the phase's mapped `data`. Each time the phase runs, the fact follows the latest result: when the result is not one of the chosen outcomes, a fact the phase published earlier in the same process — in a previous correction round, for example — is withdrawn, so an old answer is never read as a current one.

`kyb_vies` is the one document type allowed to equal an existing type: it is the type the VIES preset publishes on purpose, the same as the former KYB phase.

Proof of the query

Generates a PDF of the query and stores it with the process documents.

Field	What it does	Required	Format / values	Default	Valida messa
Generate the PDF proof	Turns the certificate on or off. Its other settings are kept while it is off.	—	On or off	Off	—
Layout	**None (standard layout)**: a table of the **Fields to print**. **VIES (VAT check)**: *The VIES layout prints the VAT check exactly as the KYB step did.*	—	One of the two	**None (standard layout)**	—

Field	What it does	Required	Format / values	Default	Valida messa
	— it reads `data.vies`, as the VIES preset produces it.				
Title (optional)	The title printed on the certificate. Empty: *Consultation evidence* (standard) or *VIES VAT registration check* (VIES).	No	Up to 120 characters, one line	—	*The pro title car have at most … charact*
Fields to print (optional)	*One path of the result data per line; add a label after a vertical bar, for example vies.name | Nome.* Without a label, the path is printed as the label.	No	Up to 30 lines. A path is up to 8 names of letters, digits, `_` and `-` joined by dots; a label up to 80 characters.	—	*The pro field "… not a va path (le digits, hyphen unders separat by dots its label too long The pro can prin most … fields.*
File name (optional)	The certificate's file name. *Letters, digits, dot, hyphen and*	No	Up to 120 characters	consulta-{documentType}-{runId}	*The file name u charact or*

Field	What it does	Required	Format / values	Default	Valida messa
	underscore, plus the placeholders {documentType}, {runId}, {nodeId} and {data.*path*} (a field of the answer).				*placeho that are allowed*

The certificate is produced only for an **answer** — Success or Not found — never for an Error, and a certificate from an earlier execution is withdrawn when the phase now ends in Error. It states the connection's name and host, the method, the time of the query (UTC), the number of attempts, the HTTP status, the result, the request identifier (the `Idempotency-Key`, when sent) and the fields you chose. It never contains request headers, credentials, the query string or the raw response body. A Not found printed with the standard layout says the record was not found instead of listing fields.

The certificate is added to the documents of the process and sent to the document management area when your company has it. Without **Publish the result**, it is filed under the type `rest_call`. If the PDF cannot be produced or stored, the phase still completes with its answer; the failure is recorded under `restCall.` node id `.evidence` in its output.

Related

- REST integration
- Overview and VIES tutorial — checks before the call
- Schemas and mapping — what `data` contains
- Cross-check — use a published fact
- Messages

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

REST integration — messages

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Every message the REST integration can show in the workflow editor, with its cause and the fix. Where a message carries a name or a number, it is shown here as `…`.

Content status	From source; not observed (`workflow-fe` `f88c5c1`, `workflow-be` `6703373`, 2026-10-22)

Where they appear. Problems of the settings are listed above the tabs as you edit, whatever tab is open. Mapping issues appear in the form of the rule they belong to, and the mapping tab counts them in *Issues to fix: …*. The other messages appear next to the field or button concerned. The editor does not block saving while problems are listed; publishing is refused by the service for the same reasons (see Refused by the service).

Messages in the editor

Message	Where	Cause	What to do
Choose a connection.	Problems	No **Connection** selected	Select a connection in **Request**.
Could not load the connections.	**Request**	The list of connections did not load	Reload the editor; if it persists, contact support.
Fill in the path.	Problems	**Path** is empty	Type the path, starting with `/`.

Message	Where	Cause	What to do
The path must be relative to the connection: no scheme or host.	Problems	**Path** starts with a scheme such as `https:` or with `//`	Keep only the path; the host belongs to the connection's **Base URL**.
The path parameter … has no rule.	Problems	The path has a name in curly braces with no entry under **Path parameters**	Edit the path again so the entry is created, then give it a source.
The path parameter … needs a source.	Problems	The entry has no **Source** and no first transform that produces a value	Pick a **Source**, or start its transforms with `constant`, `concat` or `coalesce`.
Every query parameter needs a name.	Problems	A query parameter has an empty **Name**	Name it, or **Remove** it.
The query parameter … needs a source or a fixed value.	Problems	Neither **Source**, a value-producing first transform, nor **Fixed value**	Give it one of them.
The query parameter … looks like a credential: it cannot be set in the phase and is not saved. Use a connection of type API key in the query.	Problems	The name is one of the refused credential names (see Request)	Remove it; create an **API key** connection with **API key location** **Query string**.
Every header needs a name.	Problems	A header has an empty **Header name**	Name it, or **Remove** it.
The header … cannot be set in the phase:	Problems	The name is `Authorization`,	Remove it; the connection supplies

Message	Where	Cause	What to do
authentication and secrets belong to the connection.		`Proxy-Authorization`, `Cookie`, `X-API-Key`, `Api-Key` or `Idempotency-Key`	authentication, and the Idempotency-Key option sends that header.
The timeout must be between … and … seconds.	Problems	**Timeout (seconds)** outside 1–120	Enter a whole number in range, not above the connection's timeout.
The timeout cannot be longer than the connection's (… seconds).	Problems	**Timeout (seconds)** is above the selected connection's timeout	Lower it, or raise the connection's **Timeout (seconds, 1 to 120)**.
Retries must be between … and ….	Problems	**Retries** outside 0–10	Enter a whole number in range.
The wait between retries must be between … and … seconds.	Problems	**Wait between retries (seconds)** outside 1–600	Enter a whole number in range.
Enter at least one success status code.	Problems	**Success status codes** is empty or has no number	Enter at least one 2xx code, for example `200`.
… is not a success status code: success codes must be between 200 and 299.	Problems	A success code outside 200–299	Keep only 2xx codes; use **Not-found status codes (4xx)** in **Result** for a not-found status.
… is not a 4xx status code: not-found codes must be	Problems	A code outside 400–499 in **Not-found status codes (4xx)**	Correct it, or empty the field.

Message	Where	Cause	What to do
between 400 and 499.			
Rule … needs a field in the response.	Problems	A rule of **Rules on the response body** has no **Field in the response**	Fill it, or remove the rule.
Rule … needs a value for its operator.	Problems	The operator compares with a value and **Value** is empty	Fill **Value**, or choose `exists` or `isempty`.
At most … result rules are allowed.	Problems	More than 20 rules on the response body	Remove rules.
Check … needs a field in the request body.	Problems	A check before the call has no **Field in the request body**	Fill it, or remove the check.
Check … needs a value for its operator.	Problems	Same as for result rules, on a check before the call	Fill **Value**.
The result data of check … must be a JSON object.	Problems, and under the check	**Result data (JSON object, optional)** is not valid JSON, or is not an object	Write an object, or empty the field. Until fixed, the data is not saved.
At most … checks before the call are allowed.	Problems	More than 20 checks	Remove checks.
The document type must start with a lowercase letter and use only lowercase	Problems	**Document type** is empty or malformed while **Publish the result** is on	Use a name such as `customer_lookup`, or turn publication off.

Message	Where	Cause	What to do
letters, digits and underscores (2 to 64 characters).			
The display name can have at most … characters.	Problems	**Display name (optional)** longer than 120 characters	Shorten it.
Choose at least one result to publish.	Problems	No outcome checked under **Publish when the result is**	Check at least **Success**.
The proof title can have at most … characters.	Problems	**Title (optional)** longer than 120 characters	Shorten it.
The proof field "…" is not a valid path (letters, digits, hyphen and underscore, separated by dots) or its label is too long.	Problems	A line of **Fields to print (optional)** has a bad path or a label over 80 characters	Use `path` or `path | label`, for example `vies.name | Name`.
The proof can print at most … fields.	Problems	More than 30 lines in **Fields to print (optional)**	Remove lines.
The file name uses characters or placeholders that are not allowed.	Problems	**File name (optional)** has characters other than letters, digits, `.`, `-`, `_`, an unknown placeholder, or more than 120 characters	Use only those characters and the placeholders listed under the field.
Unknown transform "…".	Rule form	The configuration names a transform	Remove the transform.

Message	Where	Cause	What to do
		type that does not exist	
The rule has no target path.	Rule form	**Target path** is empty	Type the target, or remove the rule.
Target "..." does not exist in the schema.	Rule form	The target is not a field of the **Request schema**	Correct the target, or add the field to the schema.
A source is required unless the first transform is constant, concat or coalesce.	Rule form	No **Source**, and the first transform does not produce a value	Pick a source, or start with one of those transforms.
date_format needs both the input and output formats.	Rule form	An empty format in a `date_format`	Fill both, or use `iso`.
lookup needs at least one entry in the table.	Rule form	The `lookup` table is empty	**Add pair**.
The lookup fallback is not valid.	Rule form	A stored fallback that is none of the four choices	Choose a fallback in **When the value is not in the table**.
concat needs at least one source path.	Rule form	**Source paths (one per line)** is empty in a `concat`	Add paths.
coalesce needs at least one source path.	Rule form	Same, in a `coalesce`	Add paths.
to_boolean needs at least one true value and one false value.	Rule form	A list of `to_boolean` is empty	Fill both lists.

Message	Where	Cause	What to do
The decimal separator must be "." or ",".	Rule form	A stored `number_format` with another separator	Choose `.` or `,`.
regex_extract needs a pattern of 1 to … characters.	Rule form	The expression is empty or longer than 256 characters	Write an expression up to 256 characters.
regex_extract: the group must be a whole number from 0 up to the number of groups in the pattern.	Rule form	**Capture group (0 = whole match)** is negative, not whole, or larger than the number of groups	Use 0 for the whole match, or the number of a group of the expression.
Required field "…" has no rule.	Rule list	A field the **Request schema** requires has no rule	**Map** it, or remove it from `required` in the schema.
Arrays can be nested up to … levels.	Rule form	Item rules nested more than 3 levels deep	Flatten the structure.
Item rules only apply to array targets (ending in []).	Rule form	Item rules on a target that is not a list	Make the target end in `[]`, or remove the item rules.
Source "…" was not found in the sample.	Rule form (warning)	The source is not a path of the **Input example**	Check the spelling. If the example is just incomplete, the rule still works.
Source "…" is not valid: use $form…., $process…., $doc……., $doc….[]…. or $run.id,	Rule form	A source starts with `$` but is not a well-formed reserved source: unknown root, missing key, `$doc`	Correct it, for example `$form.nif`.

Message	Where	Cause	What to do
$run.workflowId, $run.tenantId.		without type or field, unknown `$run` key	
Source "..." is not valid here: a response rule can only read what was sent, with $request.....	Rule form	A response rule reads `$form`, `$process`, `$doc`, `$run` or another `$` root	Read the answer, or `$request.` + a field of what was sent.
Source "..." reads the run ($form, $process, $doc, $run) and can only be used in top-level rules, not in item rules.	Rule form	A reserved source inside item rules	Map it in a top-level rule instead.
Invalid JSON: ...	**Schemas**	The schema, the example to infer from or the **Input example (JSON)** is not valid JSON	Fix the JSON; the message gives the parser's explanation.
Not a valid schema: describe an object (type "object") or an array (type "array").	**Schemas**	Valid JSON that does not describe an object or a list	Paste a schema whose root is an object or an array.
Save the workflow before loading an example from a run.	**Schemas**	The workflow was never saved	Save it.
No recent run reached this phase yet.	**Schemas**	No process completed the previous phase yet	Run a process, or type an example.
Loading from a run is not available in this	**Schemas**	The service does not offer it here	Type an example.

Message	Where	Cause	What to do
environment yet.			
Could not load the recent runs.	**Schemas**	Any other failure	Try again.
Choose a connection and a valid path before testing.	**Test**	No connection, empty or absolute path, or a refused header or query parameter	Fix those problems.
There is no input example: mappings will read an empty input.	**Test**	**Input example (JSON)** is empty	Add an example in **Schemas** for a meaningful test.
Too many tests in a short time. Wait a minute and try again.	**Test**	More than 10 tests per minute in your company	Wait.
The test call is not available in this environment yet.	**Test**	The service does not offer the test here	—
The test could not be run. Check the settings and try again.	**Test**	The service refused the settings, or could not make the call	See Refused by the service.
Fill in this parameter.	Preset form	A required preset parameter is empty	Fill it.
Use a two-letter country code in capitals, for example PT.	Preset form	A country parameter is not two letters	Type two letters.

Message	Where	Cause	What to do
This source is not valid: use $form...., $process.... or $doc.....	Preset form	A source parameter starts with `$` but is malformed	Correct it, for example `$form.nif`.
This connector needs a connection that does not exist yet in your organization, and only an administrator can create it. Ask an administrator to add this preset once; after that you can use it.	Preset form	First use of the preset in your company, by someone who is not an administrator	Ask an administrator to add the preset once.
The value of "..." was refused. Check it and try again.	Preset form	The service refused that parameter — for example **Requester country** without **Requester VAT number**, or the reverse	Correct it.
One of the parameters was refused. Check them and try again.	Preset form	The service refused a parameter without naming it	Check every parameter.
This preset is no longer available. Reload the page to update the list.	Preset form	The preset was removed from the catalog	Reload the page.

Message	Where	Cause	What to do
Could not prepare the connector. Try again in a moment.	Preset form	Any other failure	Try again.
The preset catalog is not available right now, so its parameters cannot be reopened. The node keeps working as configured.	Preset banner	The catalog did not load, or no longer has this preset	Try later; the phase still runs.
…: fill in the parameters to finish this connector.	Preset banner	The phase came from a workflow template and its preset parameters were never applied	**Preset parameters** › **Apply parameters**.

The messages of the **Connections** screen are in Overview and VIES tutorial.

Refused by the service

When you publish, and when you test, the service checks the configuration again. The editor shows the service's message, which names the phase and the field. These are the checks the editor cannot make while you edit:

Rule the editor does not check	What to do
The connection must exist in your company	Select another connection. A phase that came from a workflow template names the connection it expects; create a connection with that name and select it.
A header cannot be the one that carries the connection's API key	Remove the header.

Rule the editor does not check	What to do
Document type cannot be the name of one of your extraction templates	Choose another name.
Schemas must be draft-07 or 2020-12, within 256 KB, 64 levels and 20000 nodes, without `patternProperties`	Simplify the schema.
A transform's parameters must be usable: a regular expression the service can compile, a date pattern and a time zone it recognises, no value in both lists of `to_boolean`	Correct the transform.
At most 50 path or query parameters, 50 headers, 500 rules per mapping, 20 REST integration phases per workflow	Reduce them.

Errors at run time

A failure during a process is not shown in the editor: it is written in the phase's output, under `restCall.` node id `.error`, and the process continues on **Error** (or fails, when **Error** is not connected). The **Test** tab shows the same output.

`error.kind`	Meaning	Retried?
`transient`	5xx, 408 or 429; no answer within the timeout; host not found; connection failure; the credentials could not be decrypted at that moment; a **Transient (retry)** rule matched	Yes, up to **Retries**
`permanent`	Any other 3xx or 4xx; a body larger than 5 MB; an invalid configuration; the OAuth2 token request refused; a **Permanent (error)** rule matched	No
`request_schema`	The request could not be built: a mapping or a transform failed, a path parameter had no value, or the body did not match the **Request schema**	No

`error.kind`	Meaning	Retried?
`response_schema`	A success body that is not JSON, does not match the **Response schema**, or on which the **Response mapping** failed	No
`connection`	No connection configured, the connection was deleted, or its stored credentials are unreadable — enter them again in **Connections**	No
`ssrf`	The request would leave the connection's host or base path, or the host resolves to a private address	No
`not_found`	A not-found result while **Not found** is not connected	No
`preflight`	A check before the call decided **Permanent (error)**	No

`error.message` explains the cause; for a failed transform it names the rule, the transform and the value. `error.bodyExcerpt` holds up to 2 KB of the destination's answer, without the connection's secrets.

Related

- REST integration
- Request · Schemas and mapping · Result, Test and Publication
- Overview and VIES tutorial

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Workflow administration screens

Reference of the Workflow screens an administrator uses, in the order of the **Administration** and **Management** menus. Each entry states who sees the screen; the journeys explain how to use them. Screens shared with reviewers (**Human Review**, the review dossier, **Awaiting requester correction**, **Processes**) are described in the user manual's *Workflow screens* chapter. On 2026-10-22 every section of this page had its field details checked against the Workflow interface source (`f88c5c1`) and, for publication, versions, resend and permissions, the Workflow service source (`6703373`); none of it was walked in a signed-in session.

Where the screens are

Menu	Screen	Who sees it
Operations › Build AI	Build AI	Administrator
Management › Overview, **Processing time**, **Operator ranking**	Management	Administrator
Administration › Flow Configuration	Workflows and the workflow editor	Administrator, technical
Administration › Review Queues	Review queues	Administrator
Administration › Integrations & Deliveries	Webhook Deliveries	Administrator, technical
Administration › Connections	Connections	Read: every profile that sees Administration. Write: administrator.

Menu	Screen	Who sees it
Administration › Signer profiles	Signer profiles	Read: every profile that sees Administration. Write: administrator.

The menus belong to the Cogneris shell; this page describes the screens they open.

Workflows

How to get here	**Administration › Flow Configuration**
Who sees it	Administrator, technical. The service refuses creating and deleting a workflow to every other profile.
Content status	Not observed. Field details checked against the Workflow interface source (`f88c5c1`) on 2026-10-22; not observed.

Title **Workflows**. One card per workflow: name, **Draft** / **Published**, description (**—** when empty), *v… · …* (version and creation date and time).

There is no duplicate, import or export, no search and no sorting.

Buttons and actions

Button	What happens	Who can use it	Reversible?
+ New Workflow	Opens the Create Workflow dialog. Opening the screen with `?action=new` in the address opens it too.	Administrator, technical (checked by the service when you create)	—
Start	Only on a **Published** workflow. Opens the Start Workflow dialog.	Everyone who sees the screen	No — the process starts

Button	What happens	Who can use it	Reversible?
Edit	Opens the workflow editor	Everyone who sees the screen	—
Trash icon (**Delete**)	Asks *Delete "…"?* with **Cancel** and **Delete**, then removes the workflow and reloads the list	Administrator, technical (checked by the service)	No

Deleting removes the workflow from the list and its live phases, so it can no longer be started. The service keeps the deleted record; there is no way to restore it from the screen.

Common problems

Message	Cause	What to do
Loading…	The list is loading	Wait
No workflows yet. Click '+ New Workflow' to create your first.	The company has no workflow	Create one
A generic error of the request, such as *Request failed with status code 403*, instead of the cards	The list could not be loaded, or your profile is refused	Reload the page; if it persists, check your profile
The same kind of error after **Delete** — the cards disappear	The delete was refused or failed. The error replaces the list until the page is reloaded.	Reload the page; check whether the workflow is still there
Start is missing	The workflow is a **Draft** — it was never published	Publish it in the editor

Create Workflow dialog

How to get here	**+ New Workflow** on Workflows
Content status	From source; not observed (Workflow interface `f88c5c1`, 2026-10-22).

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Name	The workflow's name on the cards and in the editor	Yes	Free text; spaces at the ends are removed	Empty	*Name is required.*
Description	Shown on the card	No	Free text	Empty	—
Start from	The first content of the draft	Yes	**Blank canvas** — *Start from an empty graph and add the phases yourself.*; or one of the four templates below	**Blank canvas**	—

Template	Its description	Phases after *Start*, in order
Document intake with review	*Classify and extract incoming documents, then send them for human review before delivery.*	Classification, Extraction, Manual Review, Send

Template	Its description	Phases after *Start*, in order
Identity verification	*Crop the identity document, match the selfie against it and cross-check the extracted fields.*	Crop, Facematch, Cross-check, Manual Review
Business onboarding	*Validate the company, read its documents, cross-check the data and notify the requester.*	VIES connector (a REST integration from the VIES preset, labelled KYB, waiting for its parameters), Classification, Extraction, Cross-check, Manual Review, Send
Contract signature	*Extract the agreed terms, generate the contract, collect the signatures and notify the parties.*	Extraction, Generate document, Signature, Send

A template only lays out the phases in a line; every phase still has to be configured. The draft records which template and template version it came from.

Buttons and actions

Button	What happens	Reversible?
Create	Creates the workflow as a **Draft** and opens it in the editor — on the blank canvas with a *Start* phase, or with the template's phases in **Guided** mode	The workflow can be deleted
Cancel, or a click outside	Closes the dialog without creating anything	Yes

Common problems

Message	Cause	What to do
Name is required.	**Name** is empty	Fill it in

Message	Cause	What to do
A message from the service, or a generic error of the request	The service refused the creation — for example your profile is neither administrator nor technical	Check your profile

Start Workflow dialog

How to get here	**Start** on a published workflow in Workflows
Content status	From source; not observed (Workflow interface `f88c5c1`, 2026-10-22).

The dialog names the workflow under its title **Start Workflow**.

Fields

Field	What it does	Required	Format / values	Default	Validation message
Documents to process	The documents of the new process	Yes, at least one	*PDF, images (PNG/JPG/TIFF/BMP), DOC/DOCX, XLS/XLSX, CSV or TXT — up to 50 MB each.* You can pick files several times; the same file (name and size) is added once. Each file has a × to remove it.	None	*Select at least one document.*

The screen does not check the 50 MB limit; the service does.

Buttons and actions

Button	What happens	Reversible?
Start (reads **Starting…**)	Uploads the documents, starts a process on the live version and opens **Processes**	No
Cancel, or a click outside	Closes the dialog and forgets the files. Not possible while starting.	Yes

Common problems

Message	Cause	What to do
Select at least one document.	No file was chosen, or the upload failed without a message	Add a file; if it persists, try again later
A message from the service, or a generic error of the request	The service refused the start — for example a file too large, or a workflow that is no longer published	Check the files and the workflow's status

Workflow editor

How to get here	**Edit** on a card in Workflows, or **Create** in the Create Workflow dialog
Who sees it	Administrator, technical
Content status	Not observed. Field details checked against the Workflow interface source (`f88c5c1`) and the Workflow service source (`6703373`) on 2026-10-22; not observed.

The editor fills the whole window. Toolbar, left to right: **← Back**, the workflow name, the *… · v…* badge (**Draft**, **Published** or **Published · unpublished changes**, and the version number), **Version history**, **Editing mode** (**Guided** / **Canvas**), messages, **Undo** / **Redo**, **Save Draft**, **Publish**.

- **Nodes** palette (Canvas) or **Steps** list (Guided) — the phases; see Design and publish a workflow.
- Canvas controls: **Zoom in**, **Zoom out**, **Fit to screen**, **Reset view**.
- **Phase Configuration** panel — **Description** and the phase's settings, or **Config (JSON)** for phases without a form. **Drag to resize** widens it.
- **Version history** — every published version with **Inspect** and **Restore**.

The badge keeps the status the service last reported. Saving a draft of a published workflow leaves it published: the live version keeps running until you publish again, and while the draft differs from the live version the badge reads **Published · unpublished changes**. Publishing clears it.

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Workflow name (toolbar)	Renames the workflow; stored with **Save Draft** or **Publish**	Yes	Free text	The saved name	Refused by the service when empty, shown as a generic error of the request
Editing mode	Switches between the **Guided** step list and the **Canvas**	—	**Guided**, **Canvas**	**Canvas**; **Guided** for a draft just created from a template	—
Step type (Guided)	The phase that **Add step**	—	Every phase except *Start*, in	**Classification**	—

Field	What it does	Required	Format / values	Default	Validation messages
	appends at the end		the palette's order		
Description (Phase Configuration)	The phase's name on the canvas and in the step list	No — left empty, the previous name is kept	Free text	The phase's name	—
Config (JSON) (Phase Configuration)	The raw settings of a phase type the editor has no form for. AI Executor and Send Message show *This phase cannot run yet* instead. Start, Split and Zero Shot read no settings, so they show *This phase has nothing to*	No	Valid JSON, or empty	Empty	*This is not valid JSON, so it was not saved. Fix it or clear the field.*

Field	What it does	Required	Format / values	Default	Validation messages
	configure. instead				

The settings of every other phase are documented with the phase: see Phases.

Canvas

Element	What it does
Nodes palette	One entry per phase. Select an entry to add the phase, or drag it onto the canvas. *Start* is disabled while the flow already has one.
Pre-configured group	Below the phases, when the preset catalog loads: connector presets with a **preset** badge. Adding one opens its parameters form first; see the VIES preset.
Phase tile	Shows the phase's description and its type in capitals, with a **preset** badge for a phase made from a preset. Select it to open **Phase Configuration**; its × (**Delete phase**) removes it with its connections.
Outputs	Drag from an output at the bottom of a tile to another tile to connect them. Phases with several outputs label them: **True** / **False** (Conditional), **OK** / **NOK** (Signature validation), **Success** / **Not found** / **Error** (REST integration), **Normal** / **Alternative** (Classification). A link drawn before this version on the **Alternative** output of Extraction or Manual Review keeps its dot, with the badge *Alternative not routed*.
Zoom pill	**Zoom out** (–), the zoom level, **Zoom in** (+), **Fit to screen**, **Reset view** (back to 100%)

A connection is refused silently when it links a phase to itself, repeats an existing one, or adds a second input to a phase that takes one input only (Classification, Extraction, Manual Review, AI Executor, Send Message, Conditional, Facematch, Signature validation). *Start* has no input.

Keyboard: **Delete** or **Backspace** removes the selected phase or connection (not while typing in a field); **Ctrl**/**Cmd**+**Z** undoes; **Ctrl**/**Cmd**+**Shift**+**Z** or **Ctrl**+**Y** redoes. The *Start* phase cannot be deleted: its tile has no **×** and **Delete** / **Backspace** leave it in place.

Guided

Title **Steps**: *Build the flow as an ordered list of steps and publish it without opening the canvas.* Each step shows its number, icon and description; selecting it opens **Phase Configuration** beside the list.

Button	What happens
↑ (*Move … earlier*), ↓ (*Move … later*)	Moves the step. *Start* stays first and cannot move.
✕ (*Remove …*)	Removes the step and reconnects the line. Not offered for *Start*.
Add step	Appends the **Step type** at the end

Guided mode only shows a flow that is a single line from *Start*. A branch, a merge, a loop or a phase left unconnected shows *This flow branches, so it cannot be shown as a list. Open the canvas to keep editing it.* with **Open the canvas**. In a line, a Signature validation continues on **OK** and a REST integration on **Success**; their other outputs have to be connected on the canvas. Guided edits reorganize the canvas layout as a line.

Phase Configuration

The panel header shows the phase's icon, **Phase Configuration**, the phase type as stored (for example `REST_CALL`) and × to close. The left edge (**Drag to resize**) widens the panel by dragging, or with the left and right arrow keys. A legacy KYB phase shows the conversion banner described in KYB; a *Send* phase offers a test address and its inspector, described in Send.

Buttons and actions

Button	What happens	Who can use it	Reversible?
← Back	Returns to **Workflows**. Unsaved changes are lost without a warning.	Everyone who sees the editor	—
Version history	Opens Version history	Everyone who sees the editor	—
Undo (↶), **Redo** (↷)	Undo or redo the last change to the flow, including a deleted phase with its settings	Everyone who sees the editor	Yes
Save Draft	Stores the name and the flow. The toolbar then shows **Save Draft** in green as confirmation.	Decided by the service	Yes, by saving again
Publish	Runs the editor's checks, then asks **Publish workflow** — *Confirm publish of version …?* (the next version number) — with **Cancel** and **Publish**. Confirming saves the draft, then publishes it; the badge turns **Published** with the new number and the toolbar shows **Published**.	Administrator, technical (checked by the service)	No — but an earlier version can be restored and published again

What a publish does, from the service: it refuses the flow if any check below fails, without touching the live version; otherwise it rewrites the live phases, raises the version number by one — even when nothing changed — and stores an unchangeable copy of that version. A *Send* phase with an endpoint and no signing secret gets one at that moment. Processes already running keep the version they started on.

Common problems

Message	Cause	What to do
Loading…	The workflow is loading	Wait

Message	Cause	What to do
Workflow not found.	The workflow does not exist any more	Go **Back**
A generic error of the request instead of the editor	The workflow could not be loaded	Go **Back** and open it again
A generic error of the request after **Save Draft**	The save was refused — for example the name is empty	Fill in the name and save again
Any message under Publication refusals	The publish was refused	Fix the flow and publish again
This flow branches, so it cannot be shown as a list. Open the canvas to keep editing it.	Guided mode cannot show this flow	**Open the canvas**
A connection does not appear	It would link a phase to itself, repeat a connection, or give a single-input phase a second input	Change the connection

Version history

How to get here	**Version history** in the workflow editor
Content status	From source; not observed (Workflow interface `f88c5c1`, Workflow service `6703373`, 2026-10-22).

A panel titled **Version history**, newest first. Each version shows *v…*, **live** on the version that is running, who published it, when, *… phases* and the first 12 characters of its content fingerprint (the full value in the tooltip).

Buttons and actions

Button	What happens	Who can use it	Reversible?
Inspect	Shows *Version …, as published*, the workflow's name at that version and its phases in order	Everyone who sees the editor	—
Restore → Copy into draft	**Restore version …?** — *This copies version … into the draft. It does not change which version is live, and version … itself is left untouched. You still have to publish the draft for the restored flow to take effect.* The draft takes that version's name, description and flow, and the editor reloads it. **Cancel**, or a click outside, closes the question.	Decided by the service	Yes, by restoring another version
× (**Close version history**)	Closes the panel	Everyone who sees the editor	—

After a restore a published workflow stays published: **Start** stays in Workflows and new processes keep running the live version until you publish the restored draft. The badge reads **Published · unpublished changes** meanwhile. A workflow that was never published stays **Draft**. Restoring replaces the flow on screen: unsaved edits are lost.

Common problems

Message	Cause	What to do
This workflow has not been published yet, so it has no version history.	No version was ever published	Publish one
The version history could not be loaded. Try again in a moment.	The list or the inspected version could not be loaded	Try again later

Message	Cause	What to do
The version could not be restored. Try again in a moment.	The restore failed	Try again later

Publication refusals

Checked by the editor before publishing:

Message	Cause	What to do
Save the workflow first before publishing.	Never saved	Select **Save Draft** first
The workflow must have at least one start node.	No *Start* phase	Add one from the palette
The workflow must have just one start node.	More than one *Start* phase	Delete the extra ones
This workflow still has a legacy KYB node (…). Convert it to the VIES connector before publishing: open the node and use the conversion.	The flow still has the removed KYB phase	Convert it; see KYB
Signature validation "…": the … output is not connected.	An **OK** or **NOK** output of a *Signature validation* phase leads nowhere	Connect both outputs on the canvas

The two *start node* checks also run on **Save Draft**.

Returned by the service. The editor translates only the refusals of the *Signature validation* and KYB phases; every other refusal is shown as a generic error of the request, such as *Request failed with status code 422*, not as the explanation below:

Refusal	Cause	Shown as
A *Send* phase endpoint is not valid (`https_required`: *Send phase webhook endpoint must be a valid HTTPS URL.*)	A *Send* phase has an endpoint that is not an absolute HTTPS address (its own test address is accepted)	A generic error of the request
A *Send* phase endpoint is on another port (`port_not_allowed`: *Send phase webhook endpoint must use the standard HTTPS port 443.*)	A *Send* phase has an HTTPS endpoint on a port other than 443, which delivery would refuse	A generic error of the request
`rest_call_config_invalid`, naming the phase and the field	A *REST integration* phase is incomplete or invalid — for example no connection, or a mapping issue listed under **Issues to fix**; also more than 20 REST integrations in one workflow. See REST integration messages.	A generic error of the request
`rest_call_fact_types_unavailable`	A REST integration publishes a fact and the company's document types could not be checked at that moment	A generic error of the request; try again shortly
Workflow not found	The workflow was deleted, or belongs to another company	A generic error of the request

Refusal	Cause	Shown as
Access denied	Your profile is neither administrator nor technical	A generic error of the request
Signature validation "…": choose an existing signer profile. / *Signature validation "…": the signer profile has no signers.* / *Signature validation "…": the rule does not fit the signer profile.*	The phase has no profile, or one that was deleted; the profile is empty; or the rule asks for more signers than the profile has, or for required signers when it has none	The message itself
The KYB message above	The service found a KYB phase the editor did not	The message itself

Failed to publish workflow. appears only when the failure carries no message at all.

Starting a process on a workflow that is not published is refused; **Start** is only offered on published workflows.

Review queues

How to get here	**Administration › Review Queues**
Who sees it	Administrator. The service refuses the queues to every other profile.
Content status	Not observed. Field details checked against the Workflow interface source (`f88c5c1`) on 2026-10-22; not observed.

Title **Review queues**: *Pick a queue and choose who reviews it. A reviewer with no queue keeps seeing everything; once you assign someone, they see only their queues.* One chip per

queue with the number of documents waiting; the selected queue lists the company's members to tick, with **Find a person**, **Save**, **Cancel** and **Refresh**. See Organize human review.

The chips are the queues the service reports plus one per document type of the company's templates, even with nothing waiting; the special queues — such as **Unidentified** and **Fraud review** — come last. The first chip is selected when the screen opens. Under the selected queue's name, the members of the company are listed by e-mail, each with name and e-mail and a box ticked when the person is assigned. At the bottom: *… reviewer(s) assigned* and the note *Anyone not assigned to a queue keeps seeing every queue, including All.*

Fields

Field	What it does	Required	Format / values	Default
Queue chips	Select the queue to configure	—	One chip per queue, with its waiting count	The first queue
Find a person (*Name or e-mail…*)	Shows only the members whose name or e-mail contains the text, ignoring case	No	Free text	Empty
Member boxes	Assign or remove the person from the selected queue	No	Box	Ticked for the people already assigned

Changing queue drops unsaved ticks of the previous one without warning.

Buttons and actions

Button	What happens	Who can use it	Reversible?
Save (reads **Saving…**)	Replaces the queue's members with the ticked people and shows *Saved*.	Administrator	Yes, by ticking again

Button	What happens	Who can use it	Reversible?
	Enabled only after a change.		
Cancel	Restores the ticks to what is saved. Enabled only after a change.	Administrator	—
Refresh	Loads the queues, the templates and the members again	Administrator	—

Common problems

Message	Cause	What to do
Loading queues…	The screen is loading	Wait
Review queues repeated as a red line, and nothing else	Your profile is not administrator	Ask an administrator
This queue's document type is no longer in the catalog and nothing is waiting in it. Its assignments no longer route any document — clear them or rename the type back.	The template behind the queue was renamed or removed	Untick everyone and **Save**, or rename the type back
Nobody matches this search.	No member matches **Find a person** — or the members could not be loaded, which shows the same message	Clear the search; if the list stays empty, **Refresh**
No queue to configure yet.	There is no queue and no template	Create a template, then **Refresh**
Failed to load the review queues. or a message from the service	The queues could not be loaded	**Refresh**

Message	Cause	What to do
Failed to save the queue members. or a message from the service	The save was refused or failed	Try again

A queue missing for a new template means the template list could not be loaded; the screen does not say so. Select **Refresh**.

Webhook Deliveries

How to get here	**Administration › Integrations & Deliveries**
Who sees it	Administrator, technical. The service refuses the list, the detail and the resend to every other profile.
Content status	Not observed. Field details checked against the Workflow interface source (`f88c5c1`) and the Workflow service source (`6703373`) on 2026-10-22; not observed.

Filters **From**, **To**, **Workflow**, **Status**; columns **Date/Time**, **Workflow**, **Document**, **Endpoint**, **HTTP Status**, **Response time**, **Outcome**, **Attempt**; **Delivery detail** with **Run ID**, **Error message**, **Payload sent** and **Resend webhook**. See Connect your systems.

One row per call a *Send* phase made to your endpoint, newest first. The screen shows only the first page the service returns and has no paging: narrow the dates to reach older deliveries.

Fields

Field	What it does	Required	Format / values	Default
From, **To**	Limit the deliveries to this period	No	Date and time, in your browser's time zone	Empty — no limit
Workflow	Shows the deliveries of one	No	**All workflows** or one of the company's	**All workflows**

Field	What it does	Required	Format / values	Default
	workflow		workflows	
Status	Shows only successful or failed deliveries	No	**All statuses**, **Success**, **Failed**	**All statuses**

The filters apply only with **Apply**. The screen does not check that **From** is before **To**.

Columns

Column	Content
Date/Time	When the call was made; select it to open, or close, the **Delivery detail**
Workflow	The workflow's name, or its identifier when the name is missing
Document	The document the call carried; **—** for none
Endpoint	The address called, cut to fit
HTTP Status	The status your endpoint answered; **—** when it did not answer
Response time	In milliseconds
Outcome	**Success** or **Failed**
Attempt	The attempt number for that document of that run

Delivery detail repeats these values — **Date/Time**, **Workflow**, **Document**, **Endpoint**, **HTTP Status**, **Response time**, **Outcome**, **attempt** — and adds **Run ID**, **Error message** (only when there is one) and **Payload sent**, the body as sent, laid out as JSON.

Buttons and actions

Button	What happens	Who can use it	Reversible?
Apply	Loads the deliveries with the filters	Administrator, technical	—
Clear	Empties the filters and loads again	Administrator, technical	—
Refresh	Loads again with the filters already applied	Administrator, technical	—
Resend webhook (reads **Resending…**)	Sends the stored **Payload sent** again, once, to the stored endpoint, signed with the phase's current signing secret. The call is recorded as a new delivery with the next attempt number for that document; the list reloads, while the open detail keeps showing the delivery you resent.	Administrator, technical	No — your endpoint receives the call again
×	Closes the detail	Everyone who sees the screen	—

Common problems

Message	Cause	What to do
Loading…	The list or the detail is loading	Wait
No webhook deliveries recorded yet.	No delivery matches the filters, or none was ever made	**Clear** the filters
Unable to load webhook deliveries., or	The list could not be loaded, or your profile is refused	**Refresh**; check your profile

Message	Cause	What to do
a generic error of the request		
Unable to load delivery detail., or a generic error of the request	The detail could not be loaded	Select the row again
Webhook resent successfully.	Your endpoint accepted the new call	—
Resend failed.	Your endpoint refused or did not answer, or the service could not resend (the delivery or its stored endpoint is no longer valid)	Look at the new row's **HTTP Status** and **Error message**; fix the receiving side and resend
Workflow lists only **All workflows**	The workflows could not be loaded	Reload the page

Connections

How to get here	**Administration › Connections**, or **Manage connections** in a REST integration phase
Who sees it	Read: every profile that sees Administration. Write: administrator — the service refuses creating, changing, testing and deleting to every other profile.
Content status	Not observed. Field details checked against the Workflow interface source (`f88c5c1`) on 2026-10-22; not observed.

Title **Connections**: *Reusable base URLs and credentials for REST integrations. Secrets are write-only: once saved they are never shown again.* Columns **Name**, **Base URL**, **Authentication**, **Usage**; **New connection**, **Edit**, **Delete**, **Test connection**. See Connect your systems.

The full list of columns, form fields, buttons and messages is in REST integration › Connections. In short:

Button	What happens	Who can use it
Refresh	Loads the list again	Everyone who sees the screen
New connection, **Edit**	Open the connection form; **Save** (**Saving…**) or **Cancel**	Administrator
Test connection (**Testing…**)	While editing a saved connection, sends one GET to the base URL	Administrator
Delete	Asks *Delete this connection?*; refused while a published workflow uses it, with the list of workflows and **Dismiss**	Administrator

Main messages: *Loading connections…*, *No connections yet.*, *Could not load the connections.*, *Could not load this connection.*, *Could not save the connection.*, *Could not delete the connection.*, *The connection test could not be run.*

Signer profiles

How to get here	**Administration › Signer profiles**, or **Manage profiles** in a *Signature validation* phase
Who sees it	Read: every profile that sees Administration. Write: administrator — the service refuses creating, changing, deleting and reading a document to every other profile.
Content status	Not observed. Field details checked against the Workflow interface source (`f88c5c1`) and the Workflow service source (`6703373`) on 2026-10-22; not observed.

Title **Signer profiles**: *Who may sign the documents checked by the Signature validation phase. Signers are matched by the CPF or CNPJ in their ICP-Brasil certificate, and identifiers*

are always shown masked. Columns **Name** (with the description under it), **Signers** (*No signers*, *1 signer*, *… signers*), **Usage** (*Not used in any workflow*, *Used in 1 workflow*, *Used in … workflows*); **New profile**, **Edit**, **Delete**. See Validate digital signatures.

Earlier versions of this page marked the screen *Not available yet*. The Workflow service source (`6703373`) now contains the signer-profile service and the *Signature validation* phase; whether a given environment runs that release has not been observed.

Fields

New profile opens *New signer profile*; **Edit** opens *Edit signer profile*, above the list.

Field	What it does	Required	Format / values	Default	Validation messages
Profile name	The name chosen in the phase's **Signer profile** list	Yes	Free text	Empty	*Enter a profile name.*
Description	Shown under the name in the list	No	Free text	Empty	—

Field	What it does	Required	Format / values	Default	Validation messages
Signers	*Each signer is identified by a CPF (person) or a CNPJ (company). A stored identifier is never shown again; use Replace to change it.* One block per signer (*Signer …*).	Yes, at least one	—	None	*Add at least one signer.*
Identifier type	Whether the identifier is a CPF or a CNPJ. Locked once the signer is saved or read from a document.	Yes	**CPF (person)**, **CNPJ (company)**	**CPF (person)**	—
CPF or CNPJ	The signer's identifier. A saved or read one shows masked, with **Replace**.	Yes	Digits; punctuation is ignored. Check digits are verified.	Empty	*Enter the CPF or CNPJ.* · *This CPF or CNPJ is not valid: the check digits do not match.*

Field	What it does	Required	Format / values	Default	Validation messages
Name (for reference only)	A name to recognise the signer; not used for matching	No	Free text	Empty	—
Role	Whether the signer must sign under the *All required signers* rule	Yes	**Required**, **Optional**	**Required**	—
Certificate type	The ICP-Brasil certificate type accepted	No	**Any type**, **A1**, **A3**	**Any type**	—
Allowed issuing authorities	Accepts only certificates from these authorities	No	Names separated by commas; *Empty means any trusted ICP-Brasil authority*	Empty	—
Learn from a document (PDF)	Reads the ICP-Brasil signatures of a PDF and adds one signer per CPF or CNPJ found,	No	One PDF	—	See below

Field	What it does	Required	Format / values	Default	Validation messages
	marked *Read from the document*, with the issuer filled in				

The messages appear only after a first **Save profile**.

Buttons and actions

Button	What happens	Who can use it	Reversible?
Refresh	Loads the list again	Everyone who sees the screen	—
New profile	Opens the empty form	Administrator	—
Edit	Loads the profile into the form	Administrator	—
Add signer	Adds an empty signer block	Administrator	Yes
× (*Remove signer …*)	Removes the signer block	Administrator	Yes, until saved
Replace	Empties the masked identifier so you can type a new one	Administrator	—
Save profile	Checks the form, stores the profile and reloads the list	Administrator	Yes, by editing again
Cancel	Closes the form without saving	Administrator	Yes

Button	What happens	Who can use it	Reversible?
Delete → **Delete**	Asks *Delete this profile?* in the row, with **Cancel**	Administrator	No

Common problems

Message	Cause	What to do
Loading signer profiles…	The list is loading	Wait
There are no signer profiles yet.	None was created	Create one
Enter a profile name., *Add at least one signer.*, *Enter the CPF or CNPJ.*	The form is incomplete	Fill in what is missing
This CPF or CNPJ is not valid: the check digits do not match.	The identifier is mistyped	Correct it
Reading the certificates in the document…	The PDF is being read	Wait
Signers added from the document: …. Review the role of each one.	Signers were found and added as **Required**	Set each one's **Role**
No ICP-Brasil signature with a CPF or CNPJ was found in the document.	The PDF has no usable signature	Add the signers by hand
The document could not be read.	The PDF was refused or could not be read	Use another PDF, or add the signers by hand
This profile is used by published workflows and cannot be deleted: followed by the workflows, with **Dismiss**	The profile is in use	Change or unpublish those workflows first

Message	Cause	What to do
Could not load the signer profiles., *Could not load the profile.*, *Could not save the profile.*, *Could not delete the profile.*, or a message from the service	The request was refused or failed	Try again
New profile, **Edit** and **Delete** are missing	Your profile is not administrator	Ask an administrator

Management

How to get here	**Management › Overview**, **Processing time**, **Operator ranking**
Who sees it	Administrator. The service refuses the reports to every other profile.
Content status	Not observed. Field details checked against the Workflow interface source (`f88c5c1`) on 2026-10-22; not observed.

Reports for administrators. The service refuses them to every other profile.

Screen	What it shows
Overview	Everything open right now: totals **In progress** and **Awaiting review**, and per process type **Running**, **Awaiting review** and **Oldest**. A type opens its runs on **Processes**.
Processing time	For the last 7, 30 or 90 days, per process type: **Average**, **95th percentile**, **Slowest**, **Fastest** and how many completed; and the runs in progress with **Requester**, **Elapsed**, **Remaining**, **Expected**, flagged *Later than usual* when overdue.
Operator ranking	For the same window, documents **Decided** per operator, with **Approved**, **Rejected** and **Share**. The ranking is by volume.

A process type is a workflow; a workflow without a name shows *Unnamed process*. Durations read as minutes (*… min*), hours (*… h*) or both (*… h … min*).

Overview

Everything open right now, by type of process. Pick a type to see its runs. Each type is a card with its name, its number in progress and **Running**, **Awaiting review** and **Oldest** (the age of its oldest open run). Selecting a card opens **Processes** filtered on that workflow.

Processing time

How long each type of process takes, and when the open runs should close. Recalculated every time this screen is opened. Under the title: *Recalculated at …*. One chip per process type (**Process types**), busiest first; the first is selected. For the selected type: *… completed in the window*, **Average**, **95th percentile**, and **Slowest** and **Fastest** with the run's identifier. *… in progress* lists the open runs:

Column	Content
Requester	The first letters of the requester, then …, with the full identifier in the tooltip; **—** for a run started without a requester
Elapsed	Time since the run started
Remaining	Expected time left; *Later than usual* when the run is overdue; **—** when nothing completed in the window
Expected	Expected completion date and time; **—** when overdue or when nothing completed in the window

Operator ranking

Documents decided per operator in the window. Approvals and rejections are shown for context — the ranking is by volume. Columns **#**, **Operator** (the e-mail), **Decided**, **Approved**, **Rejected**, **Share** (a bar and the percentage of all decisions in the window).

Fields

Field	Screen	What it does	Format / values	Default
Window	**Processing time**, **Operator ranking**	The period of the report; changing it reloads	**Last 7 days**, **Last 30 days**, **Last 90 days**	**Last 30 days**
Process type chips	**Processing time**	Select the type to show; the choice is kept on reload while the type is still there	One chip per type	The first, busiest type

Buttons and actions

Button	Screen	What happens
Refresh	All three	Loads the report again
A type's card	**Overview**	Opens **Processes** for that workflow

Common problems

Message	Screen	Cause	What to do
Loading…	All three	The report is loading	Wait
Nothing is in progress right now.	**Overview**	No open run	—
No process ran in this window.	**Processing time**	No run in the window	Choose a longer **Window**
Nobody reviewed a document in this window.	**Operator ranking**	No decision in the window	Choose a longer **Window**

Message	Screen	Cause	What to do
Failed to load the report., or a generic error of the request	All three	The report could not be loaded, or your profile is refused	**Refresh**; check your profile

Build AI

How to get here	**Operations › Build AI**
Who sees it	Administrator
Content status	Not observed. Field details checked against the Workflow interface source (`f88c5c1`) on 2026-10-22; not observed.

Administrators only. Describe a process in plain words and the assistant drafts what it needs — forms, document classifiers, workflows and Word templates — through the steps **Describe**, **Questions**, **Plan** and **Done**. Nothing is created until you select **Approve and create**.

The intro reads *Describe your process and the AI drafts what it needs: forms, document classifiers, workflows and Word templates. Nothing is created until you approve.* The steps run on the service: the page address keeps the session, so you can leave and come back to it, and the screen checks progress every five seconds while it is open.

Fields

Describe

Field	What it does	Required	Format / values	Default
How does your process work?	The description the AI works from. *Who sends what, which documents you*	Yes	Free text, up to 6000 characters	Empty

Field	What it does	Required	Format / values	Default
	require, what you check and which document closes the process.			
Industry (optional)	Context for the AI	No	Up to 64 characters	Empty
Country code (e.g. PT, BR)	The country of the process; it also decides whether a company check is possible	Yes	Two letters; turned to capitals	Empty
Process language (e.g. pt-PT, en)	The language of what is drafted	Yes	Up to 20 characters	`pt-PT` when the interface is in Portuguese, otherwise the interface language
Documents you require	The documents you already know you need. Type a **Document name** and select **Add document** (or press Enter); × (*Remove …*) removes one. The same name twice is added once.	No	Up to 120 characters each	None

Continue stays disabled until the description, a two-letter country and a language are filled in.

Questions — *Before the plan*

Question	Values	Default	Shown when
Do you need a company (KYB) check?	**Yes**, **No**	**No**	The country has a source; otherwise *Company checks are only available in the European Union (VIES). There is no source for this country.*
Do you want a selfie for face matching?	**Yes**, **No**	**No**	Always
May the AI set the main rules according to market best practice? It researches the web and cites every source.	**Yes**, **No**	**Yes**	Always
Is there a final document?	**Yes**, **No**	**No**	Always
Does the final document need a signature?	**Yes**, **No**	**No**	When the service makes it depend on the previous answer, only after **Yes**
Who signs it?	Free text, up to 200 characters	Empty	After **Yes** to the signature

Then **Questions from the AI**: each with its reason, answered **Yes** / **No**, from a list, or as text up to 500 characters; all optional. *Documents the AI found in your description* offers each document as *+ …* to add it to your list.

Plan

Field	What it does	Format / values	Default
What should change?	The adjustment you ask for, for example *e.g. remove the*	Up to 4000 characters; disabled while the AI works	Empty

Field	What it does	Format / values	Default
	property tax bill; the income rule is 30%	and when no adjustment is left	
Example documents — per document type	*Each document type needs 3 examples to train its classifier. Use the Cogneris library when it covers the type, or upload your own.* A badge shows *Cogneris library · …*, *Your documents · …* or, in red, *Needs 3 examples*.	*3 to 10 files: PDF, JPG, PNG or WebP. Photos are reduced automatically.* Up to 20 MB each.	None

What the plan shows

… · Version … (the plan's title and version), then: **Resolve before approving** (each issue with where it is and why), **Forms** (with their document checklist), **Document types** (*… fields*), **Workflows** (*… steps*, with their **Rules**, each rule's reason and the web sources it cites), **Word templates**, and **Market best practice consulted** with its **Sources**. While the AI works: *Researching market best practice on the web…*, *Drafting the plan. This can take a few minutes; you can leave this page and come back.* or *Adjusting the plan…*.

Buttons and actions

Button	Step	What happens	Reversible?
Continue	Describe	Opens a session and moves to **Questions**	—
Save answers and draft the plan	Questions	Stores the answers and asks for the plan	—
Use the Cogneris library	Plan	Uses Cogneris' examples for the document type. Shown when the library covers the type and it does not use the library already.	Yes, by uploading your own

Button	Step	What happens	Reversible?
Upload your documents	Plan	Uploads the chosen files as the type's examples; *Uploading … of …* while it works	Yes, by uploading again
Adjust the plan	Plan	Sends **What should change?** and drafts a new version; *Adjustments left: …* counts what remains	—
Approve and create (reads **Creating…**)	Plan	Approves this version and starts creating everything. Enabled only when the service says the version can be approved.	No — what is created stays
Draft the plan again	Plan	Shown when there is no plan and plan drafts are left	—

Done shows **Creating your process**, then one of *Creating forms, classifiers and workflows…*, *Training the document classifiers…*, *The creation stopped with an error.*, *Training a classifier failed.*, *The status cannot be confirmed right now.*, with *Current phase: …*, until *Your process is ready to use.*

Common problems

Message	Cause	What to do
Loading the session…	The session is loading	Wait
Only tenant administrators can use Build AI.	Your profile is not administrator	Ask an administrator
The AI could not suggest more questions right now. You can continue with the questions above.	The AI questions failed	Continue without them
Market best practice could not be consulted now, so the plan only has your	The web research failed or found	Add the rules yourself with

Message	Cause	What to do
own rules. / The research found no practice with a source, so the plan only has your own rules.	nothing citable	**Adjust the plan**
This version cannot be approved yet. Adjust it or resolve the issues above.	The plan has issues	Adjust it
Choose at least 3 files., Choose at most 10 files., Each file can be up to 20 MB.	The chosen examples do not fit	Choose 3 to 10 files of up to 20 MB
A file could not be read. Use a PDF or a clear photo of the document.	The service refused an example	Use a PDF or a clearer photo
The Cogneris library is unavailable right now. Try again or upload your documents. / The Cogneris library has no examples for this document type. Upload your own.	The library could not supply examples	Upload your own
Your plan has no credits left for this step.	Your Cogneris plan has no credits left	Contact your account manager
Too many requests. Wait a moment and try again.	Too many requests in a short time	Wait and try again
The AI service is unavailable. Try again in a moment. / The AI took too long to answer. Try again.	The AI did not answer	Try again
This session reached its limit of plans. / This session reached its limit of adjustments.	The session's limits are used up	Approve a version, or start a new session
This plan cannot be approved yet.	The service refused the approval	Adjust the plan

Message	Cause	What to do
Another process is being created in this account. Try again when it finishes.	Another approval is still creating	Wait until it finishes
The plan changed meanwhile. Reload the page and try again.	A newer version exists	Reload the page
This document type is no longer in the plan.	The type was removed by an adjustment	Reload the page
This session was not found.	The session in the address does not exist	Open **Build AI** again
Some information is missing or invalid.	The service refused the input	Check the fields
Could not reach the server. Check your connection and try again.	No connection	Check your connection
Something went wrong. Try again.	Any other failure	Try again

Not observed yet

- No screen of this chapter has been walked in a signed-in session; captures are pending. Every section is documented from the Workflow interface source (`f88c5c1`) and, where stated, the Workflow service source (`6703373`).
- Behaviour decided only by the service and not visible in either source — for example the page size of **Webhook Deliveries** and the exact limits of a **Build AI** session — is not documented.
- Whether the environment you use runs the release of the Workflow service that stores signer profiles and runs the *Signature validation* phase has not been checked.

Last reviewed 2026-10-09 · Version 1.1.0 · Owner: product-documentation

Vision administration overview

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Vision is the document-AI product of Cogneris. In Vision you define **templates** — the document types your company receives and the fields to read from each — try them on real documents, tune the AI agent that extracts them, and manage the keys and webhooks that let your own systems call the Cogneris API. Workflow phases such as *Classification*, *Extraction* and *Cross-check* use what you configure here.

How to get here	Open the Vision product from the Cogneris platform. Embedded in the platform, Vision's menu appears in the platform's left column and the platform shows one breadcrumb bar; opened on its own, Vision shows its own menu (branded **Cogneris Vision**), its own breadcrumb starting at **Vision**, and a **Log out** button.
Who can use it	Any signed-in member of your company can open Vision. Templates additionally require Vision access on your membership (see Who can use what).
Content status	From source; not observed (`vision-fe` `0b724a0` , `vision-be` `fce7fcc` , 2026-10-22)

What you manage in Vision

You want to	Menu	Chapter
See how many documents and pages Vision processed in the last 30 days	**Dashboard**	Dashboard
Create a template from sample documents, follow its provisioning, rename or delete it	**Templates**	Templates — list and create
Define the fields a template extracts, test a field's prompt and publish the template	**Templates** › a template › **Schema**	Template schema
Decide what is masked in a template's documents	**Templates** › a template › **Redaction**	Template redaction
Choose how the AI agent extracts a template: mode, models, retries, domain agents, budget, run history	**Templates** › a template › **Agent**	Template agent
Try classification, extraction, split, crop and redaction on a document	**Playground**	Playground
Connect an editor (VS Code, Cursor, Claude Code) to the Cogneris MCP server	**MCP**	MCP
Set company-wide guardrails for the AI agent, read its analytics, review fraud alerts	**Tenant agent settings** (**Guards**, **Analytics**, **Review**)	Tenant agent settings
Create API keys, register webhook endpoints, read API and webhook logs	**Developers** › **API Keys**, **Webhook Endpoints**, **API Request Logs**, **Webhook Request Logs**	Developers

Menu and navigation

The menu has six entries, in this order:

Menu entry	Opens	Notes
Dashboard	The Vision home	Also where Vision opens when no page is named.
Templates	The template list	A template opens in the template editor, on its **Schema** tab. The editor's other tabs are **Redaction** and **Agent**; **Agent** has the sub-tabs **Mode**, **Router**, **Retry**, **Domain agents**, **Budget** and **History**.
Playground	The test bench for one document	—
MCP	Editor connection instructions	The former in-app Chat was replaced by this page; an old Chat link opens **MCP**.
Tenant agent settings	Company-wide agent settings	Opens on the **Guards** tab; the other tabs are **Analytics** and **Review**.
Developers	A group of four pages	**API Keys** (also what opens for the group itself), **Webhook Endpoints**, **API Request Logs**, **Webhook Request Logs**.

Every page has an address of its own, so a link to a page (for example a template's **Agent** › **Budget** tab) opens that page directly. Addresses of pages that used to live in the platform keep working.

Sign-in. Vision has no sign-in page of its own. Opening any Vision page without a session sends you to the Cogneris sign-in and brings you back to the page you asked for.

A page that does not exist shows *404* and **Page not found**, with a **Go home** button that opens the **Dashboard**.

Who can use what

Vision itself does not hide menu entries by profile: every signed-in member sees the whole menu. What you may do is checked by the Vision service on each request.

Area	Checked by the service
Templates (create, list, open, save, rename, delete, add or remove sample documents)	Your company membership must be **active** and must include **access to Vision**. The check is made live on each request, so a revoked access takes effect at once; if the check cannot be made (for example the billing service is unavailable), the request is refused.
Templates › **Redaction** (changing a redaction policy)	A company administrator. See Template redaction.
Dashboard usage figures	A signed-in member of the company. The **Active Templates** tile reads the template list, so it is subject to the Templates check above.
Other areas	Each chapter states its own checks in its **Who can use it** row.

Everything you see in Vision belongs to your company only: templates, usage, keys and logs are always scoped to your company.

How Vision relates to Workflow

Workflow runs your processes; Vision does the document work for some of its phases. What you configure in Vision changes what those phases do:

Workflow phase	What it takes from Vision
Classification	The document type, and the template of your company that covers that type. Only templates that were saved and provisioned take part in classification; see *Statuses* in Templates — list and create. When your company has no provisioned template at all, classification falls back to zero-shot classification.

Workflow phase	What it takes from Vision
Extraction	The fields declared in the template that classification found (see Template schema).
Split and Crops	The Vision service finds the segments of a file and the regions to cut. You can try split and crop on a document in the Playground.
Cross-check	The template list, to choose a document role and to suggest the template's field names. The list includes templates in every status, including **Draft**; see *What a Draft means elsewhere* in Templates — list and create.

Classification resolves the template by its **name**: the document type returned by the classifier is matched to a template whose title equals it (ignoring case, and treating spaces, hyphens and underscores alike). Keep template titles stable once Workflow processes depend on them.

How Vision relates to the Cogneris API

Your own systems reach the same document work through the Cogneris API, authenticated with the API keys created in **Developers** › **API Keys**, and they are told about results through the endpoints registered in **Developers** › **Webhook Endpoints**. The API only accepts templates that are published (status **Finished**). See Developers.

Where to start

1. **Look at the Dashboard** to see whether Vision is already processing documents for your company.
2. **Create a template** for each document type you receive — Templates — list and create — then **define its fields and save it** in Template schema. Saving provisions the template; until then it is a **Draft** and takes no part in classification.
3. **Try it** on a real document in the Playground.
4. Optionally, **tune extraction** per template in Template agent and **mask sensitive data** in Template redaction; set company-wide limits in Tenant agent settings.
5. **Use it**: in a Workflow, add Classification and Extraction phases; from your own systems, create an API key and a webhook endpoint in Developers. To build against the API from an editor, see MCP.

Related

- Dashboard
- Templates — list and create
- Developers
- Workflow phases: Classification, Extraction, Cross-check

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Dashboard

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

The Vision home: how much Vision processed for your company in the last 30 days, which document types it classified, how many templates you have, and shortcuts to the main areas.

How to get here	**Dashboard** in the Vision menu. Vision also opens here when no page is named, and **Go home** on the *Page not found* page leads here.
Who can use it	Any signed-in member of your company. The **Active Templates** tile also needs Vision access on your membership — see *Who can use what* in the Vision administration overview; without it that tile shows its error state.
Content status	From source; not observed (`vision-fe` `0b724a0` , `vision-be` `fce7fcc` , 2026-10-22)

The page is read-only. It loads two things independently when it opens — the usage figures and the template count — so one can fail while the other shows. It does not refresh on its own; reopen the page to see newer figures.

What is counted

The usage figures come from the Vision service's usage records for your company, for a window of **30 days**: today (UTC) and the 29 days before it, up to the moment the page is opened.

- **Each usage record counts as one document.** A record is written per AI operation Vision performs — for example a classification, a zero-shot classification, an extraction, a crop or a split. A single document that goes through classification and then extraction therefore counts **twice**. This is why the chart is titled *Operations & Pages*.

- **Pages** is the sum of the pages recorded on those operations.
- **Not counted:** the records of webhook deliveries and of API notification calls.
- Days are calendar days in **UTC**. Around midnight, a document processed late in the evening in a time zone behind UTC is shown on the next day.

KPI tiles

Tile	What it shows	Source
Documents — last 30 days	Number of operations in the window (see What is counted).	Usage records
Pages — last 30 days	Pages processed by those operations.	Usage records
Active Templates	Number of templates your company can use: those in **Finished** or **Training**.	The template list

Active Templates counts only templates that are ready to use — **Finished**, and **Training** (on the Google Document AI engine a template is usable while it trains). **Draft**, **Queued**, **Processing** and **Error** templates are not counted.

While a tile loads it shows …. When its data cannot be read the tile shows —, the message *Couldn't load* and a **Retry** button; the tile never shows a misleading *0* when the figure is unknown.

Charts

Operations & Pages — last 30 days

A line chart with one point per day of the window and two lines, **Documents** and **Pages**. Days without activity are drawn as zero, so the horizontal axis always covers all 30 days, labelled day/month. Pointing at a day shows both values for that day. The legend under the chart toggles a line on and off.

By document type

A donut chart of the operations that were **classifications by your trained templates**, grouped by the document type they returned, largest first. The legend beside the chart lists each type with its share of the total; pointing at a slice shows the type, its count and its share.

- Only classification against your templates is counted here. Extraction, split, crop and zero-shot classification do not record a document type, so they are left out rather than shown as one large "unknown" slice.
- A classification record with no type is shown as `unknown`.
- Document types are shown as the classifier returned them, which is normally a template title.

Chart states

State	What you see
Loading	*Loading…*
No activity in the window	*No data for the last 30 days yet.* — for the line chart when every day is zero, for the donut when there is no classification by template.
Data could not be read	*Couldn't load chart data.* and a **Retry** button.

Platform features

Under the heading **Platform features**, four cards open the main areas of Vision:

Card	Text on the card	Opens
Templates	*Create and manage AI extraction templates with configurable fields.*	Templates
Playground	*Test document extraction and classification in real time.*	Playground
MCP	*Build with the Cogneris API from VS Code, Cursor or Claude Code.*	MCP
Tenant agent settings	*Tenant-wide guardrails, budgets and analytics for the AI agent engine.*	Tenant agent settings

Each card can be clicked, or reached with the keyboard and opened with Enter or Space.

Buttons and actions

Button	What happens	Who	Reversible?
Retry (on a usage tile or a chart)	Reads the usage figures again and redraws both charts and both usage tiles.	Any signed-in member	Read-only
Retry (on **Active Templates**)	Reads the template list again.	Members with Vision access	Read-only
A feature card	Opens that area.	Any signed-in member	—

Common problems

Message	Cause	What to do
Couldn't load on **Active Templates** only	The template list was refused or the Vision service did not answer. A membership without Vision access is refused.	Press **Retry**. If it persists, ask your company administrator to check that your membership is active and includes Vision.
Couldn't load on the two usage tiles and *Couldn't load chart data.* on both charts	The usage figures could not be read.	Press **Retry**.
No data for the last 30 days yet. although documents were processed	The documents were processed more than 30 days ago, or (for the donut only) none was classified against one of your templates.	Expected. Use the Playground or a Workflow with a Classification phase.
Documents is higher than the number of files you sent	Each operation on a document counts once — a document	Expected; see What is counted.

Message	Cause	What to do
	classified and then extracted counts twice.	

Related

- Vision administration overview
- Templates — list and create
- Developers — request and webhook logs per call

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Templates

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

A template describes one type of document your company receives — an invoice, an identity card, a proof of address — through sample documents and a list of fields to extract. This chapter covers the template list, the **New Template** screen, the **sample documents** a template learns from, and the **statuses** a template goes through until it is published and used by classification, by Workflow and by the Cogneris API. Defining the fields and saving (publishing) the template happens in the editor, described in Template schema.

How to get here	**Templates** in the Vision menu. **Add New Template** on the list opens **New Template**. The Playground also offers **Create Template** when a document matches none of your templates.
Who can use it	Members whose company membership is active and includes access to Vision. The Vision service checks this live on every template request and refuses it when the check cannot be made. See *Who can use what* in the Vision administration overview.
Content status	From source; not observed (`vision-fe` `0b724a0` , `vision-be` `fce7fcc` , 2026-10-22)

In the Vision service a template is called a **classifier**: the API paths, the service messages and the Workflow logs use that word. It is the same object.

Template list

Header **Templates**, with the subtitle *Manage and organize your extraction templates*. The list shows every template of your company, newest first, as cards.

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Search (placeholder *Search templates…*)	Keeps the cards whose title or description contains the text.	No	Free text; case is ignored. Filters as you type, in the browser.	Empty	—
Filter by Status	Keeps the cards in one status.	No	**Filter by Status** (all), **Draft**, **Processing**, **Finished**	**Filter by Status** (all)	—

The status filter has one option per status: **Draft**, **Queued**, **Processing** (also covers provisioning), **Training**, **Finished** and **Error**.

What a card shows

Element	What it shows
Thumbnail	The first page of the template's first sample document (PDF or image). When it cannot be shown — no file, or the preview link expired — a document icon is shown instead.
Title	The template's title. Click the card to open the template in the editor, on its **Schema** tab.
Pencil (**Edit name**)	Shown only for a **Draft** template. See Rename a template.
Information button	Opens a small panel with **Created by** (the user who created the template) and **Created at** (the creation date, in your browser's date format). Click it again to close it.

Element	What it shows
Status dot	One coloured dot; point at it to read the status name. Grey: **Draft**. Amber (animated): **Processing** or **Queued**. Green: **Training** or **Finished** — see Statuses for why **Training** is green. Red: **Error**.
Processing with an hourglass	Shown next to the dot while the template is being provisioned (**Processing** or **Queued**).
N file / *N files*	Number of sample documents. Hidden when there are none.
Engine mark	A small round mark, titled *Cogneris AI*, shown whenever the template records an engine — whichever engine it is.

Pointing at a card shows its action icons in the top-right corner: **Reprocess** (only when the template is stuck or failed) and **Delete template**.

Live status

While a template is being provisioned, the list follows it live: the dot turns green or red the moment the Vision service finishes, without reloading. When Vision is embedded in the platform, each status change is also announced in the platform's notification bell; the text of those notifications is the service's own, and some of it is in Portuguese. If the live connection cannot be opened, the list instead re-reads itself every 5 seconds, and only while at least one template is **Processing** or **Queued**.

Empty states

What you see	Meaning
Six grey placeholder cards	The list is loading.
Manage and organize your extraction templates and no cards	Your company has no template yet — **or the list could not be read**: the screen does not distinguish the two. Reload the page if you expect templates.
No templates found matching your search. and **Clear filters**	The search text or the status filter hides every card. **Clear filters** empties both.

Buttons and actions

Button	What happens	Who	Reversible?
Add New Template	Opens **New Template**.	Members with Vision access	—
A card	Opens the template editor on the **Schema** tab.	Members with Vision access	—
Edit name (pencil)	Turns the title into a text box.	Members with Vision access; **Draft** templates only	Yes — rename again
Information button	Shows **Created by** and **Created at**.	Members with Vision access	—
Reprocess	Starts provisioning again. While it runs the tooltip reads **Reprocessing…**. See Reprocess a template.	Members with Vision access; shown for **Processing** and **Error**	No — the template goes back to **Processing**
Delete template	Asks for confirmation, then deletes. See Delete a template.	Members with Vision access	**No**
Clear filters	Empties the search and the status filter.	Anyone on the screen	Yes

Statuses

A template is always in one of six statuses. The service decides the status; you cannot set it directly.

Status	Dot	Meaning	Takes part in classification?	Accepted by the Cogneris API?
Draft	Grey	Created, with its sample documents, but never saved in the editor. Nothing has been provisioned. Every template starts here.	No	No
Processing	Amber	Saved; the service is provisioning it — reading the samples and building what classification needs. Also the status while a sample change is being applied.	Not yet	No
Queued	Amber	Google Document AI engine only: the training request was refused because the engine's training capacity was busy. The service retries on its own every **15 minutes** until training starts.	Not yet	No
Training	Green	Google Document AI engine only: the engine has accepted the template and is importing the samples and training. The service treats the template as ready to use at this point (its	Decided by the engine while it trains	No — the API waits for **Finished**

Status	Dot	Meaning	Takes part in classification?	Accepted by the Cogneris API?
		notification says so) and finishes on its own.		
Finished	Green	Provisioned and **published**.	Yes	Yes
Error	Red	Provisioning failed — for example the samples could not be read or imported, or the engine refused the training.	No	No

From Draft to Finished

1. **Create** the template with at least one sample document on **New Template**. It is created as **Draft**, and you land in the editor.
2. **Define the fields** and **save** in the editor — see Template schema. Saving stores the field schema and, unless the template is already **Training** or **Finished**, starts provisioning: the template moves to **Processing**.
3. **Provisioning** runs in the background. Its path depends on the engine your company uses:
   - **Cogneris engine** (the default): the service reads the sample and writes the template's entry in your company's classification catalogue, then sets **Finished** — or **Error** if anything fails.
   - **Google Document AI engine**: the service creates the dataset and labels on the engine, sets **Training**, imports the samples and trains. When training cannot start because the engine is busy, the template is **Queued** and retried every 15 minutes. When training completes, **Finished**; on failure, **Error**.
4. Saving again a template that is already **Training** or **Finished** stores the new schema but does not provision again.

Which engine is used is a setting of the Vision service; the engine is recorded on the template when it is created.

Published templates and versions

A **Finished** template is **published**: its identifier is a public contract that Workflow processes and API callers rely on. The service therefore protects it:

On a Finished template	Result
Change the title	Refused: *A published classifier is immutable. Create a new classifier to publish a changed template.* The list also hides the pencil.
Add or remove a sample document	Refused. The service answers as if the template did not exist (*Classifier not found.* / *Classifier file not found.*).
Reprocess	Refused with the same *published classifier is immutable* message. The list does not offer **Reprocess** for it.
Change the field schema	**Allowed**, in place. Each change that alters the definition records a new **version** of the template (1, 2, 3 …); saving an unchanged definition records nothing.

Versions. The service keeps an append-only history of each published template's definition. Version 1 is the definition at the time it is first published or first used; each later change to the schema adds the next number. A document job submitted through the Cogneris API is pinned to the **current version at the moment it is accepted**, so a schema change made afterwards does not change how that job is extracted; the next job uses the new version. The history is kept as audit evidence even after the template is deleted. Vision does not show the history on screen.

To change the title or the sample documents of a published template, create a new template.

What a Draft means elsewhere

A template that is not **Finished** (or, on the Google engine, **Training**) is invisible to most of the product:

Where	Effect of a Draft, Processing, Queued or Error template
Classification (Workflow phase and the Playground)	The template is not in your classification catalogue, so no document is classified as that type. If your company has **no** provisioned template at all, classification falls back to zero-shot classification and the AI names the type itself.
Extraction	Extraction follows a template only when classification resolved one, so a Draft's fields are not used.
Cross-check	The **Document** list of a rule shows **all** your templates, Drafts included. A rule whose role is a Draft template finds no document classified under that template; the role also carries the template's title as document type, so it is filled only by a document whose classified type has that same title. Expect the rule to be `inconclusive` with reason `role_not_present`.
Cogneris API	A document job naming the template is refused at submission: *Template … is not available to this tenant.* The API's template list shows only **Finished** templates.
Dashboard	Counted in **Active Templates**.

Classification matches a template by its **title** (ignoring case, and treating spaces, hyphens and underscores alike). Two templates with titles that differ only in those ways are ambiguous; keep titles distinct.

Sample documents (training samples)

Sample documents are the real documents a template learns from. They are uploaded when the template is created (at least one), and can be added or removed in the editor while the template is not yet published — see Template schema.

What the service does with them:

- **They are stored for your company only.** With the Cogneris engine they are kept encrypted at rest; with the Google Document AI engine they are kept where the engine can read them directly.

- **They teach classification.** On the Cogneris engine, provisioning sends the **first** sample document to the AI and only lists the names of the others; the catalogue entry is built from that first document. On the Google engine all samples are imported and used for training.
- **They are the documents you test fields against** in the editor: field discovery and field tests run on a sample document.
- **Adding or removing a sample after provisioning does not re-provision the template.** On a template in **Error** or stuck in **Processing**, use Reprocess to provision again with the current samples. On a **Finished** template, samples cannot change.
- **Each file** is checked by the service against Vision's document upload rules (type, content and size); the maximum size is **50 MB** per file. Empty files are refused.

The **Create Template** screen does not restrict the file picker to particular types; a file the service does not accept is refused when you create the template.

New Template

Header **New Template**. Create a template from a title, a description and one or more sample documents.

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Title	The template's name. It is shown in the list, and classification uses it to match a document type to this template.	Yes	Free text; spaces, accents and punctuation are kept. Surrounding spaces are removed. Must not be the title of another template of your company. On the Google	Empty	*Please provide a template title.* — and from the service: *Classifier already exists for the subscription.*, *Template name cannot contain 'google' when the Google Document AI*

Field	What it does	Required	Format / values	Default	Validation messages
			Document AI engine it must not contain `google`. Placeholder: *Document: Invoice*.		*engine is selected. Rename the template and try again.*
Description	A short description, shown nowhere else on this screen; it is searched by the list's search box.	Yes (on this screen)	Free text. Placeholder: *Brief template description*.	Empty	*Please provide a short template description.*
Sample documents (drop zone **Drag documents here**, *or select them from your computer*)	The sample documents of the template. Drop files on the zone or use **Select file**. Each file appears in a list under the zone with a **×** button (**Remove**).	Yes — at least one	Several files at once; you can add more in several rounds. Accepted types and the 50 MB limit are checked by the service.	None	*Please select at least one document for the template.*

The checks run in the order **Title**, **Description**, files; only the first problem is shown, at the top of the form.

Buttons and actions

Button	What happens	Who	Reversible?
Select file	Opens your computer's file picker.	Members with Vision access	Yes — remove the file
× (**Remove**) next to a file	Takes the file out of the list.	Members with Vision access	Yes
Cancel	Returns to the template list. Nothing is created.	Members with Vision access	—
Create Template	Checks the fields, then uploads everything in one request. While it runs the button reads **Uploading…** and the form is locked. On success the template exists as **Draft** and the editor opens on its **Schema** tab, showing the files you just picked.	Members with Vision access	Delete the template from the list

What the service does on **Create Template**:

1. Checks the title is present and not already used by your company, and checks every file.
2. Creates the template as **Draft**, records the engine your company uses, and stores the sample documents. If storing a file fails, nothing is kept.
3. Makes sure your company has the engine resources it needs to provision templates later. This step runs **after** the template is created: if it fails you see an error, but the template already exists.
4. Records the creation in the audit trail.

Nothing is provisioned yet: the template stays **Draft** until you save it in the editor.

Arriving from the Playground

When the Playground classifies a document that matches none of your templates, it offers **Create Template**, which opens this screen with that document attached and, when a type was detected, the type as the title. Review the title and add a description.

Rename a template

Available from the list for **Draft** templates only.

1. Click the pencil (**Edit name**) next to the title. The title becomes a text box with the text selected.
2. Type the new title. **Enter** or leaving the box saves; **Esc** cancels.

An empty or unchanged title cancels the change. A title another template of your company already has is refused before anything is sent (*Another template is already named "…"*). The new title shows immediately; if the service refuses it, the old title comes back and the reason is shown above the list. On the Google engine, keep titles free of `google`. The description is kept unchanged.

Reprocess a template

Reprocess forces provisioning again for a template that is stuck or failed. The list shows it for **Processing** and **Error** templates.

- The template goes back to **Processing** and is provisioned with its current samples and schema, as when it was saved.
- If the template is **Training**, nothing is started: the engine is already working on it.
- A **Finished** template cannot be reprocessed (it is published).
- A **Queued** template is not offered **Reprocess**; the service retries it on its own every 15 minutes.

Each reprocess is recorded in the audit trail.

Delete a template

1. Point at the card and click the bin icon (**Delete template**).
2. The dialog **Confirm delete** shows the template's title. For a **Finished** template it adds: *Deleting this template before a pending job is processed may cause that job to fail if the template cannot be resolved. No other template will be used as a fallback.*
3. Click **Yes, delete**, or **Cancel** to keep it.

While it is deleted the card shows **Deleting…**; then it disappears from the list.

What the service deletes: the template, its sample documents in storage and, for a **Finished** template on the Google Document AI engine, the trained classifier on the engine. If the engine-side removal fails, nothing is deleted. The version history of a published template is

kept as audit evidence. **Deleting cannot be undone**: to use the document type again, create a new template.

Before deleting, check that no Workflow phase, review queue or API caller still relies on the template: a document job already accepted for it fails when it is processed, and no other template replaces it.

Common problems

Message	Cause	What to do
Please provide a template title. / *Please provide a short template description.* / *Please select at least one document for the template.*	A required field is empty.	Fill it in.
Could not upload the documents: … (with the service's text, for example *Classifier already exists for the subscription.*)	The service refused the request. The text says why: a title already used by your company, a refused file, a title containing `google` on the Google engine.	Fix what the text names and press **Create Template** again.
Could not upload the documents. Please try again.	The request failed without a reason from the service (network, timeout).	Try again. Before retrying, check the list: the template may have been created.
Could not retrieve the created template data.	The service answered without the created template.	Check the list; the template may exist. Open it from there.
Classifier already exists for the subscription. right after an error on the first attempt	The first attempt created the template and then failed in a later step.	Open the existing template from the list instead of creating it again.

Message	Cause	What to do
A template stays **Processing** for a long time	The provisioning job was lost or is stuck.	Press **Reprocess**.
A template is **Error**	Provisioning failed; the notification text gives the reason.	Check the sample documents (readable, the right type), then **Reprocess**. If it fails again, create the template again with other samples.
A template stays **Queued**	Google Document AI engine: the engine's training capacity is busy.	Wait; the service retries every 15 minutes.
Documents are never classified as the new template	The template is still **Draft** (never saved), or not yet **Finished**.	Open it, define the fields and save — see Template schema.
A published classifier is immutable. Create a new classifier to publish a changed template.	You tried to rename or reprocess a **Finished** template.	Create a new template.
Template … is not available to this tenant. (Cogneris API)	The API call names a template that is not **Finished**, does not exist, or belongs to another company.	Publish the template, or use the identifier of a **Finished** one.
The list is empty although templates exist	The list could not be read; the screen shows the empty state instead of an error.	Reload the page. If it persists, check your Vision access.

Message	Cause	What to do
A rename did not stick	The service refused it; the screen restores the old title silently.	Try again. A **Finished** template cannot be renamed.

Related

- Template schema — define the fields, add or remove sample documents, save and publish
- Template redaction and Template agent — the other editor tabs
- Playground — try a template on a document
- Workflow phases: Classification, Extraction, Cross-check
- Developers — API keys for the Cogneris API

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Template editor — Schema

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

The template editor is where you decide what Vision extracts from a type of document. Its **Schema** tab holds the template's sample documents on the left and its list of fields on the right: you let the AI propose the fields from a sample (**fields definition**), adjust names, types and extraction prompts, test a prompt against the sample, and save the result with **Save template**.

How to get here	**Templates** › click a template card. Creating a template on the new-template screen also opens it here. The editor opens on the **Schema** tab.
Who can use it	Every call this screen makes goes to the Vision service, which only answers members of the company whose membership is active and who have access to Vision. A lookup failure is a denial. The screen itself checks no role.
Content status	From source; not observed (`vision-fe` `0b724a0` , `vision-be` `fce7fcc` , 2026-10-22)

The editor shell

Before showing any tab, the editor checks that the template exists and belongs to your company.

- While it checks: *Loading template…*
- If the template does not exist, was deleted, or belongs to another company: the heading **Template not found** and *We couldn't find this template, or it doesn't belong to your account. Return to the templates list to pick another one.* No tab is shown.
- Otherwise, three tabs (**Template editor tabs**):

Tab	What it holds
Schema	This page.
Agent	How the agent runs extraction for this template. See Agent.
Redaction	What this template may redact. See Redaction.

Each tab has its own address (`/templates/edit/…/schema` , `…/agent` , `…/redaction`); opening the editor without a tab goes to **Schema**.

Header

Element	What it shows or does
Back (arrow)	Returns to the **Templates** list. Unsaved changes on this tab are lost without a warning.
Breadcrumb	**Templates** / the template's name (or **Templates** while the name is unknown).
Document badge	*… documents* — how many sample documents the template has. Shown when it has at least one.
Save template	Saves the schema. See Save the template.

Two banners can appear under the header:

- A red banner (errors) or a toast (successes) for the messages listed on this page. Most disappear after 4 seconds.
- When the template is already published (status **Finished**), a permanent note: *Editing this template updates it in place. Pending jobs may use the updated schema or fail if the template cannot be resolved.* See Published templates and versions.

Document Preview

The left pane, **Document Preview** — *Choose a file and define the schema from the selected sample.* — shows the template's sample documents. One is selected at a time; it is the

document that **fields definition** reads and that **Run Test** tests against.

Buttons and actions

Button	What happens	Who	Reversible?
Add document (in the pane header, and **+** in the thumbnail bar)	Opens the file picker. You can pick several images (any image type) or PDFs; each is uploaded on its own and appears at once as a thumbnail while it uploads. The first document of an empty template is selected automatically. *Document added successfully.* when the upload is confirmed.	Any Vision user	Yes — remove the document
Zoom out	Shrinks the preview by 20 %, down to about 40 %.	Any Vision user	Yes
Zoom in	Enlarges the preview by 20 %. Above 100 % you can drag the page (mouse or touch) to move around it.	Any Vision user	Yes
Fit to center	Resets zoom to 100 %, removes the rotation and centres the page. Selecting another document does the same.	Any Vision user	—
Rotate	Turns the preview 90° clockwise. Display only: the stored document is not changed.	Any Vision user	Yes
fields definition	Asks the AI to propose the template's fields from the selected document. See fields definition.	Any Vision user	No — it replaces the field list
Hide thumbnails / **Open thumbnails**	Switches the thumbnail bar between large tiles (default; a page-1 image of each document) and a compact strip of file names.	Any Vision user	Yes

Button	What happens	Who	Reversible?
Thumbnail or file name	Selects that document.	Any Vision user	—
x on a thumbnail	Asks *Are you sure you want to delete this document?* (**Confirm deletion**). **Delete** removes the document from the template and from storage; **Cancel** or Escape closes the dialog. Not available while the document is still uploading or being removed.	Any Vision user	No

When a stored PDF cannot be rendered, the preview shows *Select a document to preview.* and its tile shows the first three letters of the file name. With no document at all, the stage shows *Select a document to preview.* and the compact strip shows **No document**.

What the service does with documents

- An uploaded document goes through the service's document admission checks and file validation (at most 50 MB per file) and is stored for the template. A rejected file shows *Failed to upload document.*
- Documents can be added or removed only while the template is not published. On a **Finished** template the service refuses both, and the screen shows *Failed to upload document.* or *Failed to remove document.*
- A removed document is deleted from storage. The success message says *Document removed and template is reprocessing.*, but for a template that is not published nothing is reprocessed.

fields definition

fields definition sends the selected document to the AI and asks it to describe the document's structure — field names, types and extraction prompts, no values.

1. The button is disabled while a definition is already running, while no document is selected, while a document is being removed, and while the template is being saved. Its tooltip says why: *Extraction in progress — please wait*, *Select a ready document to define extraction fields*, or **fields definition**.

2. When it starts, the pane shows *Extracting data…* with a rotating message every 5 seconds (*Still processing*, *Identifying field structure*, *Getting document information*, *Almost there*, *Just a little more*) and the elapsed seconds.
3. When it ends, the proposed fields replace the list: *Extraction fields loaded successfully.* If it fails or returns no field, the current list stays as it was.

It also runs **on its own, once**, when the first document finishes uploading and the template has no fields yet — including right after you create a template.

What the service does:

- It reads the stored document and runs the template-discovery instruction with the engine configured for it. The AI is told to use `snake_case` names, to write short English prompts of the form "extract …", to choose a type among `string`, `number`, `date`, `boolean`, `array` and `object` (tables become `array`), to mark a field `required` only when it is evidently always present, and to include only fields it can see.
- **On a template that is not published, the proposed schema is saved on the template immediately**, before you click **Save template**. Leaving the editor does not undo it. On a published template it is not saved until you click **Save template**.
- The call is billable: it consumes credits, is recorded in usage, and is refused when the company's plan has a hard stop and its allowance is exhausted. It is also rate-limited with the other document-processing calls.

Field Settings

The right pane, **Field Settings** — *Edit field names, visibility and output types.* — lists the template's fields. Above the list, a summary: *… fields* (every field, columns of tables included), *… tested* (fields with a test result) and *avg confidence …%* (the rounded mean of the test confidences). With no field yet: *No fields loaded yet.* and *Use* `fields definition` *after selecting a document.*

There are two kinds of row:

- **A simple field**: type chip, name, the last test confidence (*…%*) when the field has been tested, the type list, **Settings** (gear) and **Remove field** (bin).
- **A table** (type **Table**) or **an object** (type **Object**): a group with a chevron (**Collapse** / **Expand**, shown when it has sub-fields), the name (placeholder **Table**), a **Table** or **Object** badge, the type list, **Settings** (gear), **Add field to table** and **Remove field**. Its columns or sub-fields are listed indented under it, each as a simple field — or as a nested group.

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Name (placeholder **Field name**, or **Table** for a table)	The key the field has in the extraction result. Typing a space inserts `_`; any character other than letters, digits and `_` is removed as you type. Changing the name refreshes the field's prompt only while it is still the default prompt of its type (or empty); a prompt you wrote is kept (see Field types).	Yes	Letters `a-z` `A-Z`, digits, `_`; must start with a letter or `_`	Empty for a new field	*Only letters (a-z, A-Z), digits, and underscores. Must start with a letter or underscore.* (simple fields; a table name is only outlined in red)
Type list (**Field Settings**)	What the field holds and how its value is post-processed. Changing the	Yes	The 14 types in Field types	**String** for a new field	—

Field	What it does	Required	Format / values	Default	Validation messages
	type refreshes the field's prompt to the default prompt of the new type only while the prompt is still a generated default (or empty); a prompt you wrote is kept. Choosing **Table** opens the group so you can add columns.				
Prompt (in the settings drawer)	The instruction the AI follows to extract this field.	No	Free text	The default prompt of the type, or the prompt proposed by **fields definition**	*Prompt is required to run the test.* (only when testing)

A field also carries two properties that this screen keeps but does not let you change: `required` (set by **fields definition** when the AI judged the field always present; `false` for fields you add) and `visible` (`true` unless a loaded schema says otherwise). The service code does not act on either; they reach the AI only as part of the schema it is given. A `description` that **fields definition** may propose is dropped on save.

Field types

The type list offers 14 types. The default prompt is what the field's prompt becomes when you set its name or type; you can then change it in **Settings**. The last column applies to every extraction with the template — Playground, synchronous API and extraction jobs; see What happens at extraction time.

Type (key)	What it extracts	Default prompt (for a field named `x`)	What the service does with the value
String (`string`)	Any text.	`Extract field x`	Returned as extracted.
Number (`number`)	A numeric value.	`Extract field x`	Rewritten as a plain number: currency symbols and letters removed, thousands separators dropped, decimal separator `.`. With both `.` and `,` present, the last one is the decimal separator; a lone `,` is decimal only with 1 or 2 digits after it, and a lone `.` with exactly 3 digits after it is read as a thousands separator. Left unchanged if it cannot be read.
Date (`date`)	A date.	`Extract field x`	Rewritten as `YYYY-MM-DD` (or `YYYY-MM` for a month and year) when it can be read — numeric formats such as `dd/MM/yyyy`, `yyyy-MM-dd`, `MM/dd/yyyy`, `yyyyMMdd`, and

Type (key)	What it extracts	Default prompt (for a field named `x`)	What the service does with the value
			month names in English or Portuguese. Left unchanged otherwise.
Table (`array`)	A list of rows — line items, a table. Its columns are the fields you add with **Add field to table**.	`Extract field x`	Returned as a list. If the AI returns comma-separated text instead, it is split into a list. Each row's columns are post-processed by their own types. A table without columns is saved as a list of text values.
Signature presence (`signature`)	Whether a signature is present.	`Extract field x. If a signature is present, return true. If no signature is found, return false.`	Returned as extracted.
Face presence (`face`)	Whether a face photo is present (for example on an ID document).	`Extract field x. If a face photo is present in the document (e.g. an ID document), return true. If no face photo is found, return false.`	Returned as extracted.
Checkbox (`checkbox`)	Whether a checkbox is ticked.	`Extract field x. If the checkbox is checked, return true. If the checkbox is not`	Returned as extracted.

Type (key)	What it extracts	Default prompt (for a field named `x`)	What the service does with the value
		`checked, return false.`	
Email (`email`)	An email address, as written.	`Extract field x. Find and return the email address exactly as written (e.g. name@example.com). If no email address is found, return null.`	Returned as extracted.
Phone number (`phone`)	A phone number with its country and area codes, as written.	`Extract field x. Find and return the phone number, preserving all country and area code digits exactly as written. If no phone number is found, return null.`	Returned as extracted.
Barcode (1D) (`barcode_1d`)	The decoded value of a one-dimensional barcode (Code 128, EAN-13, UPC…).	`Extract field x. Locate the 1D barcode (e.g. Code 128, EAN-13, UPC) and return its decoded value. If no 1D barcode is found, return null.`	Returned as extracted.
QR code (`qr_code`)	The text, URL or data embedded in a QR code.	`Extract field x. Locate the QR code, decode it, and return the embedded text, URL, or data. If no QR code is found, return null.`	Returned as extracted.

Type (key)	What it extracts	Default prompt (for a field named `x`)	What the service does with the value
Stamp / Seal (`stamp`)	Whether an official stamp or seal is present.	`Extract field x. If an official stamp or seal is present in the document, return true. If no stamp or seal is found, return false.`	Returned as extracted.

For the presence, checkbox, email, phone, barcode and QR types, what comes back is decided by the prompt and the AI model; the service applies no type-specific processing to them.

Two more types are in the list:

- **Object** (`object`) — a group of fields. It is shown as a group, like a table, and its sub-fields are listed under it; add them with **Add field to table**. Sub-fields of an object are post-processed by their own types.
- **Yes/No** (`boolean`) — a true/false value, returned as extracted.

Older schemas whose types are written in Portuguese (`lista` , `objeto` , `numero` , `data`) are read as **Table**, **Object**, **Number** and **Date**; any type the editor does not know is read as **String**.

Buttons and actions

Button	What happens	Who	Reversible?
Add field	Adds an empty **String** field at the **top** of the list. Enables **Save template**.	Any Vision user	Yes — remove it
Add field to table (on a table or object)	Adds an empty **String** column or sub-field at the top of the group and opens it.	Any Vision user	Yes — remove it

Button	What happens	Who	Reversible?
Collapse / **Expand** (chevron on a table)	Hides or shows the table's columns.	Any Vision user	Yes
Settings (gear)	Opens the prompt drawer for the field, table or object. See Prompt drawer.	Any Vision user	—
Remove field (bin)	Asks *Are you sure you want to remove the field …?* (**Remove field**). **Remove** deletes the field — a table with all its columns — from the list: *Field "…" removed.* **Cancel** or Escape keeps it.	Any Vision user	Only by leaving without saving, or by adding the field again

There is no way to reorder fields, no import or export of the schema, and no undo on this screen. Every change on this tab lives only in the browser until you click **Save template** — except the schema that **fields definition** saves on its own (see fields definition).

Prompt drawer

Settings on a field opens a drawer on the right. Its title is the field name (or **Field Prompt** when the field has none), with the field's type under it. Clicking outside the drawer, **Close** (×) or Escape closes it and discards an unsaved prompt.

Field	What it does	Required	Format / values	Default	Validation messages
Prompt	The extraction instruction for this field (placeholder *Define the extraction prompt*).	To run a test	Free text; leading and trailing spaces are removed	The field's current prompt	*Prompt is required to run the test.*
Result	Read-only. The value the last test	—	Text	Empty	—

Field	What it does	Required	Format / values	Default	Validation messages
	returned (placeholder *Run test to see result*), with **Confidence** … % next to the heading.				

Button	What happens	Who	Reversible?
Run Test (*Running test…* while it runs)	Runs the prompt currently in the box — saved or not — against the selected document and shows the value and its confidence. *Field test completed for "…".* The result and confidence also appear on the field's row and in the summary.	Any Vision user	—
Save Prompt	Keeps the prompt on the field and closes the drawer: *Prompt updated for "…".* The template itself is not saved yet.	Any Vision user	Yes — edit it again

What the service does with a test: it runs a single-field instruction with the engine configured for field tests. Only your prompt is sent — not the field's name or type — so the prompt alone must say what to extract and in which format. The AI returns one value (or nothing when the value is absent, illegible or uncertain) and a confidence from 0 to 100, which the service clamps to that range. Like **fields definition**, a test is billable, recorded in usage, and rate-limited. Test results are not stored with the template.

Tables have no **Settings** button, so a table's own prompt cannot be edited here: it keeps the prompt from **fields definition**, or the default `Extract field …` it got when you named it.

Save the template

Save template builds the schema from the field list and sends it to the service, then returns to the **Templates** list.

It is disabled — with the tooltip that explains why — while saving (*Saving template…*), while **fields definition** runs (*Wait for the extraction to finish before saving*), and while no field has been defined or added yet (*Run "fields definition" first — there is nothing to approve yet*). Otherwise its tooltip is *Save template*.

What is saved, per field:

```
{
  "invoice_number": { "type": "string", "prompt": "Extract field
invoice_number", "required": false, "visible": true },
  "items": {
    "type": "array", "prompt": "Extract field items", "required": false,
"visible": true,
    "items": { "type": "object", "properties": { "amount": { "type":
"number", "prompt": "Extract field amount", "required": false, "visible":
true } } }
  }
}
```

- Saving is refused, with a message, when a simple field has no name (*Every field needs a name before you save.*), when a name breaks the naming rule (*These field names can't be used: …*), or when two fields at the same level share a name (*These field names are used more than once at the same level: …*). A table with an empty name is still saved as `items` (then `items_2`, `items_3` …).
- A table that was switched to another type loses its columns on save.

What the service does:

Template status	Call	Effect
Not published (draft, queued, processing, error, training)	Finish	Saves the schema. Unless the template is already training, it moves to **Processing** and its provisioning job is queued; the **Templates** list follows the status until the template is **Finished**.
Finished (published)	Update	Saves the schema in place, without provisioning again, and records a new definition version if the schema changed.

The schema is stored encrypted, and each save is written to the audit trail.

Published templates and versions

A template with status **Finished** is published. On a published template:

- **Save template** updates the live schema. Each save that changes the schema records a new numbered definition version, starting at 1; saving an unchanged schema records nothing. A version is also recorded the first time an extraction reads a template that has none yet. The version history is kept by the service and is not shown on this tab.
- **fields definition** still proposes fields, but they are only saved when you click **Save template**.
- Sample documents cannot be added or removed (see What the service does with documents).

That is what the banner *Editing this template updates it in place. Pending jobs may use the updated schema or fail if the template cannot be resolved.* warns about: an extraction job that was accepted with a definition version extracts with that version; one accepted without a version reads the live schema when it runs.

What happens at extraction time

When a document is extracted with this template — in the Playground or through the API — the service turns the saved schema into the instruction the AI follows:

1. **The schema is loaded** for the caller's company and decrypted.
2. **It is embedded whole in the prompt** — every field with its type, prompt, `required` and `visible`, tables with their columns — with the instruction to return **only the top-level field names, spelled exactly**, not to invent, rename or add fields, and to return an empty value with confidence 0 for a field the document does not have. Any extra instructions of the caller are appended.
3. **The AI returns** a value and a confidence for every field (a list of rows for a table).
4. **Values are post-processed by type**: dates and numbers are normalized and tables split into lists, as described in Field types. The other types are returned as extracted. Extraction jobs apply the same step as a direct extraction, so the same document returns the same values either way.
5. A direct extraction records, with its usage, the template and the definition version it ran on.

The two paths differ when the template cannot be used:

Path	Template missing, of another company, not readable, or with no fields	Template not yet Finished
Direct extraction (Playground, synchronous API)	The extraction goes on **without** the schema, with the caller's own instructions only. The service logs a warning; the result simply has no template.	The live schema is used.
Extraction job (asynchronous)	The job fails with `template_unresolved` and is not retried.	The job fails with `template_unresolved`.

A job accepted with a definition version uses that version's schema, even if the template has been edited since; the template must still exist and belong to the company.

Common problems

Message	Cause	What to do
Template not found (heading) / *Template not found.*	The address names a template that does not exist, was deleted, or belongs to another company.	Go back to **Templates** and open the template from there.
Template id not provided.	The editor was opened without a template id.	Open the template from **Templates**.
Failed to load template.	The template could not be read.	Reload the page.
No documents available to extract fields.	The template was just created without any document.	Add a document with **Add document**.
Template not identified.	A document was picked before the template's id was known.	Reload the editor and add the document again.

Message	Cause	What to do
Failed to upload document.	The service refused the file (type, size over 50 MB, admission checks) or the template is published.	Check the file; on a published template, samples cannot change.
Could not verify document addition.	The upload answered, but the new document was not in the answer.	Reload the editor to see whether the document was added.
Failed to remove document.	The service refused the removal — the template is published, or the document is no longer there.	Reload the editor.
Please select a ready document to define extraction fields.	**fields definition** without a selected document.	Select a document.
This document is not ready for extraction yet.	The selected document is still uploading.	Wait for the upload to finish.
Could not obtain extracted fields for this document.	The AI returned no schema.	Try again, or with another sample.
The document took too long to process. Try again, or use a smaller file.	The definition timed out.	Retry, or use a smaller document.
This file format is not supported for extraction. Use a PDF, PNG or JPG.	The service did not accept the file format.	Use a PDF, PNG or JPG.

Message	Cause	What to do
This file is too large to process. Try a smaller document.	The file is too large for the definition.	Use a smaller document.
The document quality is too low to read reliably. Try a sharper scan.	The service answered with a validation error. This message is shown for **any** validation error of the call, not only for image quality.	Retry with a sharper scan; if it persists, the cause is another validation error.
Could not extract the fields from this document. Please try again.	Any other failure — including a refused call when the plan's allowance is exhausted.	Try again; check the plan's usage.
Select a field and a ready document before testing.	**Run Test** with no document selected.	Select a document.
Prompt is required to run the test.	**Run Test** with an empty prompt.	Write a prompt.
This document is not ready for testing yet.	The selected document is still uploading.	Wait for the upload to finish.
No result was returned for this field.	The AI found no value for the prompt.	Rewrite the prompt, or test on another sample.
Failed to run the selected field test.	The test call failed (including a refused call when the plan's allowance is exhausted).	Try again.
Could not identify the template to finalize.	**Save template** without a template id.	Reopen the template from **Templates**.

Message	Cause	What to do
Failed to finish the template with extracted data.	The save was refused or failed. Nothing changed on the screen; your edits are still there.	Try again.
A custom prompt was replaced by `Extract field …`	You renamed the field or changed its type after writing the prompt.	Set the name and type first, then write the prompt.
A field is missing after saving	It had no name.	Give every field a name before saving.

Related

- Templates: list and create
- Template editor — Agent
- Template editor — Redaction
- Playground — run an extraction with the template.
- Developers — call extraction through the API.
- Extraction phase — use the template in a workflow.
- Vision overview

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Redaction policy

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Each template can carry a redaction policy: the sensitive-data categories Cogneris covers when a document of this template is redacted, how sure a detection must be, how the output image is produced and how long the redacted file is kept. A policy is edited as a draft, validated against the tenant ceiling and published; a published version can never be changed.

How to get here	**Templates** › open a template › **Redaction** tab (next to **Schema** and **Agent**). See Templates.
Who can use it	Company or platform administrators. The screen opens for the roles `tenant.admin`, `cogneris-admin` and `cogneris-platform-admin` — the same roles the service accepts. Anyone else sees **Company or platform administrator access required** — *Ask a company or platform administrator to configure and publish this policy.*
Content status	From source; not observed (`vision-fe` `0b724a0` , `vision-be` `fce7fcc` , 2026-10-22)

How a policy moves

1. **Tenant ceiling first.** The company has one redaction ceiling: the categories any template may redact and the maximum retention. It is configured in Company administration, not on this screen. Without it no draft can be created: *A tenant Redaction ceiling must be configured in Company administration before a draft can be created.*
2. **Create draft.** The first draft of a template starts from the ceiling: every category the ceiling allows is selected, retention is 24 hours (or the ceiling's maximum, if lower), raster DPI is 300, no per-category confidence threshold is stored, and every category gets a

solid black box. Later drafts are a copy of the latest published version. A template has at most one unpublished draft at a time.

3. **Edit and Save draft.** Each save increases the draft's **Revision**.
4. **Validate.** The service checks the saved revision against the current ceiling.
5. **Publish.** After a confirmation, the version becomes immutable and is used for new redaction jobs of this template.

The screen header shows **Draft version …** and **Revision …** while a draft is open.

Screen states

What you see	When
Loading redaction policy…	The draft or the version history is loading.
The redaction policy could not be loaded. with **Retry**	Loading failed. The error message, if any, is shown under the title.
Create a redaction policy draft — *Start from the active tenant ceiling or the latest published version.* with **Create draft** (**Creating…** while it runs)	The template has no draft and no published version.
Published version … — *This version is immutable and is now available to Redaction workflows.* with the publication time and the first 12 characters of its content hash	No draft is open and a published version exists, or a version was just published. Buttons: **Create new draft**, **View version history** / **Hide version history**.
The editor (ceiling banner, two cards and the **Publication** rail)	A draft is open.

Tenant ceiling banner

Shown above the editor.

Element	What it shows
Tenant ceiling and the **Active** chip	The ceiling is configured. The line under it reads **Revision …** · **… categories allowed** · **… hours maximum retention**.
Warning style, no chip	No ceiling is configured: *A tenant Redaction ceiling must be configured in Company administration before a draft can be created.*

Sensitive-data categories

Select the subset this template may redact. Categories outside the tenant ceiling remain visible but cannot be selected.

The 13 categories are grouped in four columns. A category the ceiling does not allow is shown greyed and its checkbox is disabled.

Group	Category (label)	Service name
Identity and biometrics	**Full name**	`FullName`
	Date of birth	`DateOfBirth`
	Face	`Face`
	Signature	`Signature`
Government identifiers	**Tax ID**	`TaxId`
	Document number	`DocumentNumber`
	Passport	`Passport`
	QR or barcode	`MachineReadableCode`
Contact details	**Address**	`Address`
	Phone	`Phone`

Group	Category (label)	Service name
	Email	`Email`
Financial	**Bank account**	`BankAccount`
	Payment card	`PaymentCard`

When you select a category, it gets the opaque black box style, and the current shared confidence threshold if one is set. At least one category is required.

Detection and output

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Shared confidence threshold	Confidence a detection must reach to count as a confident one, applied to every selected category. Shows *Mixed* (empty field) when the selected categories have different thresholds; typing a value sets all of them.	No	Number 0–1, step 0.01	1 (no threshold stored means 1)	*Confidence must be between 0 and 1.*
Redaction style	How each region is covered. The only option is **Opaque black**	Yes	**Opaque black box**	See left	—

Field	What it does	Required	Format / values	Default	Validation messages
	box: solid black (`#000000`), fully opaque, no label, 1-pixel margin. Shows *Mixed* when any selected category has a different style. The margin does not count: the 4-pixel black box a new draft starts with reads as **Opaque black box**. Picking **Opaque black box** rewrites every selected category to it.				
Raster DPI	Resolution at which pages are rasterized before detection and covering. The output is always a raster image, never the original file with boxes on top.	Yes	Whole number 200–600	300	*Raster DPI must be a whole number between 200 and 600.*
Temporary retention	How long the redacted	Yes	Whole hours, 1	24, or the ceiling's	*Retention must be*

Field	What it does	Required	Format / values	Default	Validation messages
(hours)	artifact and its job are kept before they expire and are cleaned up. Stored in seconds; the screen shows hours.		up to the ceiling's maximum	maximum if lower	*greater than zero.* · *Retention cannot exceed the tenant ceiling.*

The service also accepts per-category thresholds and styles (colour `#RRGGBB`, opacity exactly 1, optional label `REDACTED`, margin 1–4,048 pixels). This screen sets them all at once and offers only the black box; values written in another way are kept until you change the shared control.

Other validation messages shown in the editor:

- *Select at least one category.*
- *Remove categories that are outside the tenant ceiling.*

Publication rail

Step	Done when	Text
1	No unsaved edits	**Draft saved** / **Unsaved changes**
2	The current revision passed **Validate**	**Validated against the current ceiling** / **Validation required**
3	Steps 1 and 2 are done	**Ready to publish** / **Save and validate this revision to publish**

Under the steps: *Published versions are immutable.*

Buttons and actions

Button	What happens	Who	Reversible?
Create draft / **Create new draft**	Creates a draft from the ceiling (first time) or from the latest published version.	Administrators	Yes — a draft can be edited freely and is replaced by the next one
Save draft (**Saving…**)	Checks the form, then saves it as a new revision. Enabled only when something changed. The service checks the categories and retention against the ceiling again.	Administrators	Yes
Validate	Asks the service to check the saved revision against the current ceiling. Enabled only when there are no unsaved edits. Any later edit clears the result.	Administrators	—
Publish	Opens the confirmation dialog. Enabled only when the current revision is saved and validated.	Administrators	—

Button	What happens	Who	Reversible?
Publish version … (in the dialog **Publish redaction policy?** — *Version … will become immutable after publication.*; **Publishing…** while it runs)	Publishes the version. The service locks the ceiling, re-checks the draft, records the publication time and a SHA-256 content hash, and writes an audit entry.	Administrators	**No.** A published version cannot be edited or deleted. To roll back, publish a new version with the old settings.
Cancel (in the dialog)	Closes the dialog without publishing. Also closed by Escape or by clicking outside.	—	—
View version history / **Hide version history**	Shows or hides the **Version history** card.	Administrators	—
Retry	Reloads the policy (after a load failure, or after a stale-draft error).	Administrators	—

Version history

Lists up to 100 versions of this template's policy, or *No policy versions yet.*

Column	Content
Version	Policy version number.
State	**Published** or **Draft**.
Publication time	Date and time of publication; `–` for a draft.

Column	Content
Ceiling revision	Revision of the tenant ceiling the service reports with the version; `–` when it reports none.
Content hash	First 12 characters of the SHA-256 hash of the published settings; `–` for a draft.

If the history fails to load, the card shows *The request could not be completed. Try again.* with **Retry**.

What the service does with a published policy

- **Only the latest published version is used.** When a redaction job starts for a template, the service takes its latest published version. A draft is never used. Without a published version the job is refused.
- **The ceiling is re-checked at use.** If the ceiling has since been narrowed so that the version's categories or retention no longer fit, the job is refused as outside the ceiling. Publish a new version that fits.
- **Jobs keep their policy.** Each job stores a snapshot of the policy it started with; publishing a new version does not change jobs already running.
- **Detection.** Pages are rasterized at **Raster DPI** and only the selected categories are looked for. Each detection is compared with its category's threshold: a detection below the threshold is still covered, but it is flagged low-confidence (`redaction.low_confidence`) in the job's manifest.
- **Covering and verification.** Regions are covered with the category's style. An independent check then looks for any sensitive data left; if it finds some, the regions are widened by 8 pixels and covered again, up to 6 times. If data is still found, the job fails (`redaction.residual_pii_detected`).
- **Retention.** The job and its artifact expire at the job's start time plus **Temporary retention**, and are then cleaned up.
- **Limits.** A document can have at most 100 pages and 100 MiB.
- Redaction is available only where the service's redaction feature is enabled for the tenant.

Common problems

Message	Cause	What to do
A tenant Redaction ceiling must be configured in Company administration before a draft can be created.	The company has no redaction ceiling.	Configure the ceiling in Company administration, then **Create draft**.
Company or platform administrator access is required.	The service refused your role.	Ask a company or platform administrator.
This draft changed elsewhere. Reload the latest revision before continuing.	Someone saved, validated or published this draft after you loaded it. The editor is locked.	Click **Retry** to reload the latest revision, then redo your changes.
The draft conflicts with the current tenant ceiling.	A selected category or the retention is outside the current ceiling. The affected fields are listed under the message.	Remove the listed categories or lower the retention, save and validate again.
The draft could not be validated.	**Validate** failed for another reason.	Try again; if it persists, reload.
The request could not be completed. Try again.	Any other failure.	Try again.
Publish stays disabled	The revision has unsaved edits or was not validated.	**Save draft**, then **Validate**.

Related

- Templates
- Template schema
- Template agent
- Playground
- Vision overview

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Template agent

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

The **Agent** tab configures how Cogneris refines a template's extraction after the first pass: which AI model runs each step, how often low-confidence fields are re-extracted, which domain agents (such as fraud detection) run, and how much a run may cost. You can test an unsaved configuration on a sample document and review every run of the template.

How to get here	**Templates** › open a template › **Agent** tab (between **Schema** and **Redaction**). It opens on **Mode**. See Templates.
Who can use it	Tenant administrators and technical administrators (`customer-admin` , `customer-tech`) and the Cogneris roles `cogneris-admin` , `cogneris-tech` and `cogneris-platform-admin` . The service refuses reading, saving, cost preview and dry runs for other roles. The run history is readable by any signed-in user of the tenant.
Content status	From source; not observed (`vision-fe` `0b724a0` + cogneris-vision-fe#192, `vision-be` `fce7fcc` + cogneris-vision-be#504, 2026-10-22)

How the agent works at run time

- **When it runs.** When an extraction is requested with a template and the agent engine is enabled in the service, Cogneris first extracts the document in a single pass. If the template has a **saved** agent configuration, the agent then refines that result; without one, the single-pass result is returned unchanged.
- **Checks before the run.** The saved configuration is combined with the tenant's agent guards (see Tenant agents). The run is refused, and the extraction fails with `code: reason` , when the tenant has paused agents (`agents_paused`), when a model the run could use breaks the tenant's model rules (`model_not_allowed`), or when an enabled

domain agent is not on the tenant's allowed list (`agent_disabled`). Only synchronous mode is accepted.

- **Monthly cap.** Before a run (not a dry run), the tenant's month-to-date agent spend is compared with its monthly cap; at or above the cap the run fails with `monthly_cap_exceeded`. See Budget.
- **Refinement.** A validator model scores every extracted field. Fields below the **Confidence threshold** are re-extracted by a focused extractor call and scored again, up to **Max attempts** times (see Retry). Then each enabled domain agent runs in turn.
- **Limits during the run.** The whole run must finish within the sync timeout (`sync_timeout`, see Mode), and no model call may push the run over the per-run cost cap (`budget_exceeded`, see Budget).
- **What is kept.** Every run is recorded with its status, cost, the configuration it actually ran with, and one entry per step (model, latency, cost, outcome). These appear under History.

Effective configuration banner

A collapsible bar above the tabs, labelled **Effective configuration**. Expanded, it shows as JSON what would actually run for this template: the saved configuration after the service has applied the platform's default per-run budget and the tenant guard checks. It reflects the **saved** configuration, not unsaved edits, and refreshes after each save.

- It shows an empty JSON object when the template has no saved configuration, or when the guard checks would refuse the run (the reason is not shown here — run a dry run to see it).
- After the **Router** tab has computed a cost preview, the bar also shows **≈ … per typical doc**.
- If the service cannot be reached, the bar is not shown.

Save bar

Always visible at the bottom of the tab. All six tabs edit one shared draft; nothing is stored until **Save**.

Button	What happens	Who	Reversible?
Test on sample	Opens the dry-run dialog with the current, unsaved draft. See Dry run.	Administrators	—

Button	What happens	Who	Reversible?
Discard	Returns every tab to the last loaded or saved configuration. Enabled only when there are unsaved changes.	Administrators	—
Save (**Saving…**)	Sends the whole configuration with the version number you loaded. Enabled only when there are unsaved changes.	Administrators	Yes — save again with other values

Versions. The first save stores version 1; each later save increases the number by one. The number only detects concurrent edits: if someone saved in between, your save is refused with `version_conflict: expected …, actual …. Reload and retry.` Previous configurations are not kept, so there is no version to roll back to. What a past run used is kept with the run itself.

A save keeps settings that have no control on this screen as they are stored — in particular `react_retry_enabled` (whether the retry pass runs at all; on by default), which you set through the API.

When the configuration is saved, the service also removes workflow-only agents (face extraction, facematch, cross-document validation) from the enabled list; those run only when a workflow step asks for them.

Messages that can appear in the save bar:

Message	Cause	What to do
`version_conflict: expected …, actual …. Reload and retry.`	Someone else saved this template's configuration after you opened it.	Reload the page, reapply your changes, save.

Message	Cause	What to do
`retry.max_attempts must be in [0, 4].` · `retry.confidence_threshold must be in [0.0, 1.0].` · `per_run_budget_cents must be greater than 0; omit it to use the default.` · `sync_budget must be greater than 0s and at most 60s; omit it to use the default.`	A value outside the service's range (the screen normally prevents these). A cost cap of 0 or a sync timeout of 0 seconds is refused because no run could succeed with it.	Correct the value, or leave the field empty to use the default.
Request failed	Loading or saving failed without a message from the service.	Try again.

Mode

Card **Execution mode**.

Fields

Field	What it does	Required	Format / values	Default	Validat messa
Execution	How the agent runs. Runs are always **Synchronous**; the card states it, and there is no other mode to choose.	—	**Synchronous**	**Synchronous**	—

Field	What it does	Required	Format / values	Default	Validat messa
Sync timeout (seconds)	Maximum time the whole agent run may take. The current value is shown next to the label. *Maximum time to wait for sync extraction. Range: 1–60s. Engine default is 12s when unset.*	No	Slider, whole seconds 1–60	12 (not stored until you move the slider)	`sync_bu must be greater than 0s at most 60s; omi it to us the default.`

At run time. When the time runs out, the run stops and fails with `sync_timeout` (*Sync budget of … s elapsed before the agent finished.*). The service's hint for this error suggests faster or cheaper models. A timeout of 0 seconds is refused when you save, because the run would have no time at all; a configuration saved with 0 before this check existed opens with the default (12s) and is repaired by the next save.

Router

Card **Per-step model selection** — *Choose which AI model handles extraction and validation. Tier presets fill both steps with fast, balanced, or premium models.*

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Extractor	Model pinned for the focused re-	No	**Use tenant default**, or one of the active models recommended for	**Use tenant default**	—

Field	What it does	Required	Format / values	Default	Validation messages
	extraction in Retry.		this step, shown as `name – tier` (**Fast**, **Balanced** or **Premium**)		
Validator	Model pinned for the validator that scores fields, and for the other validator-type calls of a run.	No	As above	**Use tenant default**	—

Loading… is shown while the model lists load. If they fail to load, the lists stay empty.

Buttons and actions

Button	What happens	Who	Reversible?
Tier preset: Fast / **Balanced** / **Premium**	For each of the two steps, picks the first model of that tier in the step's list. A step with no model of that tier is left as it is.	Administrators	Yes — **Discard** or change the selects

Cost preview

About 0.4 s after each change, the service estimates the cost of the configuration and the card header shows **≈ … per typical doc** in US dollars. The estimate uses fixed token counts per step (extractor 5,000 in / 2,000 out, validator 3,000 / 1,000), the model's price, and a typical document of 5 pages. Steps you have not pinned are estimated with the models the platform would use. If the estimate fails: **Preview cost failed** followed by the service's message.

At run time.

- **Use tenant default** removes the pin: the step uses the model the platform configures for it, with the platform's fallback model if the first one is rate-limited.
- A pinned model always wins and has **no fallback**.
- A run dispatches extractor and validator calls only. The screen no longer offers **Planner** or **Post-process** pins; a pin for them saved by an earlier version stays in the configuration but is neither used by a run nor priced by the preview.
- Saving does not check pins against the tenant's model rules. A pin that breaks them (provider, region, model class, no-training, zero-retention, or the health-data rule) is refused when the run starts, with `model_not_allowed`.

Retry

Card **Retry policy** — *When a field's confidence falls below the threshold, the engine re-extracts that field up to the configured number of times.*

Fields

Field	What it does	Required	Format / values	Default	Validation messa
Max attempts	How many focused re-extraction rounds a run may make. The current value is shown next to the label. *0–4 attempts.*	Yes	Whole number 0–4 (typed values are rounded and clamped)	2	`retry.max_attempts in [0, 4].`

Field	What it does	Required	Format / values	Default	Validation messa
Confidence threshold	Score a field must reach to be left alone. *0.00–1.00. Higher values trigger more retries.*	Yes	Slider 0.00–1.00, step 0.01	0.85	`retry.confidence_th` `must be in [0.0, 1.0`

At run time. The validator scores every field (this score is the validator's own, not the extractor's self-reported confidence). While any field is below the threshold and attempts remain, the run makes one focused extractor call for those fields and one more validator call. Each round therefore costs two model calls. **Max attempts** 0 keeps the scoring but never re-extracts. Fields still below the threshold at the end stay flagged; whether they go to human review is decided by the tenant's minimum field confidence (see Tenant agents), not by this threshold.

Domain agents

Card **Domain agents** — *Domain-specific agents add extra checks (fraud detection, GL coding, cross-document validation) to extraction runs for this template.*

Notice: *Per-agent settings are now delivered to enabled agents at runtime. Some agents may still ignore specific fields as their implementations mature.*

The list comes from the service and contains every registered agent that a template may enable, each with a checkbox, its name and its key. At `vision-be` `fce7fcc` these are:

Agent (label)	Key	Settings
Fraud detection in Playground	`fraud-detection`	**Fraud detection settings**: **sensitivity** (required) — select **Select**, `low`, `medium`, `high` — *How aggressive to flag potential fraud markers.*
ReAct retry on low-confidence fields	`react-retry`	None — *This agent doesn't expose a configuration schema yet.*

The settings form appears only while the agent is enabled. Agents of a workflow (face extraction, facematch, cross-document validation) are not offered here.

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Agent checkbox	Adds the agent to the template's enabled list.	No	On / off	Off	—
Fraud detection in Playground checkbox	As above. Disabled when the tenant has not allowed fraud detection, with *Fraud detection in Playground is unavailable because your tenant has not enabled it.*, or when the tenant guards could not be loaded, with *Fraud detection*	No	On / off	Off	—

Field	What it does	Required	Format / values	Default	Validation messages
	in Playground is unavailable because tenant permissions could not be loaded.				
sensitivity	Fraud risk appetite. It moves the two risk thresholds only: `low` review 0.50 / suspect 0.85; `medium` 0.35 / 0.70; `high` 0.20 / 0.55.	Marked required	`low`, `medium`, `high`	Unset, which runs as `medium`	—

Agents with other settings would render them with a generic form: text, number (with its minimum and maximum), yes/no, a choice list, or a comma-separated list (*Comma-separated list of values.*, placeholder `value-1, value-2, value-3`). Any other type shows *This field type isn't yet supported by the v1 form renderer.*

At run time. Enabled agents run after the retry pass, in the order they were enabled, each on the result of the previous one; each one's cost adds to the run, and a failing agent fails the run. Fraud detection examines the refined fields and the document's bytes and records a fraud assessment (see Playground). If the tenant's allowed list of domain agents is not empty and does not contain an enabled agent, every run fails with `agent_disabled`.

Messages: *Loading…*; **Could not load domain agents** followed by the error; *No domain agents are registered yet.*

Budget

Card **Budget**.

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Per-run cost cap (USD)	Hard cost ceiling for one run. *Optional. Hard cap per run. Leave empty to use the tenant default.*	No	US dollars, two decimals (stored in cents), at least `0.01`; a value that rounds to 0 cents or is negative is ignored	Empty (placeholder `0.00`)	`per_run_budget_cents must be greater than 0; omit it to use the default.`

At run time. Before every model call the service adds an estimate of that call to what the run has already spent; if the sum would exceed the cap, the run stops with `budget_exceeded` (*Estimated next-step cost (…¢) would exceed per-run budget (…¢, accumulated …¢).*). The service's hint: *Increase the per-run budget on the template, or move the validator to a cheaper model.* When the field is empty, the service applies the platform's default per-run budget if one is configured, otherwise no per-run cap. A cap of 0 is refused when you save, because it would refuse every model call; a configuration saved with 0 before this check existed opens with the field empty and is repaired by the next save.

Month-to-date spend

Below the field, the **Month-to-date spend** chart shows one bar per day of the current month with this template's agent spend in US dollars, and a dashed line at the monthly cap.

Element	Content
MTD spend	Sum of this template's daily spend since the first of the month. Dry runs are not counted.
Monthly cap	The tenant's monthly cost cap, which is what runs are checked against. Hidden when there is no cap.

Messages: *Loading…*; **Could not load chart** followed by the error.

Monthly cap and plan ceiling. The monthly cap is a tenant-wide setting on Tenant agents, not on this tab. When it is saved, the service checks it against the maximum cost cap of the company's billing plan (a plan maximum of 0 means unlimited) and refuses a higher value with `plan_ceiling_exceeded: requested … > ceiling …. Upgrade your plan to raise this cap.` If billing cannot be reached the save is refused (`plan_ceiling_unavailable`). The check is skipped when a Cogneris platform administrator has set an override for the tenant. At run time, a cap of 0 means no monthly cap; otherwise every non-dry run is refused once the tenant's month-to-date spend over all templates reaches the cap.

History

Card **Run history**: the runs of this template, newest first, 100 at a time.

Column	Content
Status	Run status as the service reports it, colour-coded.
Started	Start date and time.
Finished	End date and time; `–` while running.
Cost	Total cost in US dollars.
Flags	**Dry run** for runs started from **Test on sample**.
Error	Error code of a failed run, such as `budget_exceeded`, `sync_timeout`, `monthly_cap_exceeded`, `model_not_allowed` or `policy_blocked`.

Click a row to open **Run detail**: **Status**, **Started**, **Cost**, **Error** (when there is one) and **Steps** — one line per step with its kind, model, latency in ms, cost and outcome, or *No steps recorded.* Close it with **×**.

Buttons and actions

Button	What happens	Who	Reversible?
Load more (*Loading…*)	Loads the next 100 runs. Shown while more exist.	Any tenant user	—
Export CSV (**Exporting…**)	Downloads `agent-runs-…csv` with up to the 5,000 most recent runs of this template: `request_id`, `template_id`, `status`, `started_at`, `finished_at`, `total_cost_cents`, `is_dry_run`, `error_code`. A failed export only re-enables the button.	Any tenant user	—

Messages: *Loading…*; **Could not load runs** followed by the error; *No runs recorded for this template yet.*

Dry runs are deleted 30 days after they were made (the service sweeps them every hour). Other runs stay.

Dry run

Test on sample opens **Dry-run on a sample** — *Run this template's draft configuration against a sample document. The result is stored for 30 days and is not used for billing.*

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Sample document	The document to test on.	Yes	Any file	—	`sample_doc_base64 is required (FE must inline the`

Field	What it does	Required	Format / values	Default	Validation messages
	The file name and size in bytes are shown once chosen. The file is sent inline with the request.				`test document bytes).`

Buttons and actions

Button	What happens	Who	Reversible?
Run (**Running…**)	Runs the current draft — saved or not — on the sample. Enabled once a file is read.	Administrators	—
Cancel / **×**	Closes the dialog and clears it.	—	—

Result

Element	Content
Result header	**Run ID**, status (`succeeded` or `failed`) and total cost, plus the fraud check badge for this run.
Error line	`code: reason` when the run failed.
Refined draft (dry-run output)	The fields the agent produced, as JSON, or *No refined draft was produced.*

Element	Content
Production comparison	*A side-by-side production extraction will appear here in a future update.*

What the service does.

- The draft is applied over the saved configuration: its execution mode and retry settings win; its model pins are added to the saved ones (a pin removed only in the draft still applies); an empty domain-agent list or an empty per-run cap falls back to the saved value.
- The same tenant guard checks as a real run apply. A refusal shows as *Dry-run failed* or as `code: reason`.
- The monthly cap is not checked, and dry runs are left out of month-to-date spend.
- The run starts from an empty set of fields — the template's single-pass extraction is not run on the sample first — and the sample counts as one page.
- The run is recorded in History with the **Dry run** flag and deleted after 30 days.

Common problems

Message	Cause	What to do
`version_conflict: …`	Concurrent save.	Reload, reapply, save.
`agents_paused: Tenant has paused agents.`	Agents are paused for the tenant.	Resume agents on Tenant agents.
`model_not_allowed: Model '…' …`	A pinned or platform model breaks the tenant's model rules.	Pick another model on **Router**, or adjust the tenant guards.
`agent_disabled: Domain agent '…' is not in tenant allowed_domain_agents.`	An enabled agent is not allowed for the tenant.	Disable it on **Domain agents**, or allow it on Tenant agents.
`budget_exceeded`	The per-run cost cap is too low for the models used.	Raise **Per-run cost cap (USD)** or choose cheaper models.

Message	Cause	What to do
`sync_timeout`	The run took longer than **Sync timeout (seconds)**.	Raise the timeout or choose faster models.
`monthly_cap_exceeded`	The tenant's month-to-date spend reached the monthly cap.	Raise the cap on Tenant agents (within the plan) or wait for the next month.
Fraud detection in Playground is unavailable because your tenant has not enabled it.	Fraud detection is not on the tenant's allowed list.	Allow it on Tenant agents.
The effective configuration shows an empty JSON object	No saved configuration, or the run would be refused.	Save the configuration; run a dry run to see the reason.

Related

- Templates
- Template schema
- Redaction policy
- Tenant agents
- Playground
- Vision overview

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Playground

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

The Playground is where you try Vision on a real document before you put it in a workflow or an API integration. On one screen you can extract a document against your templates (**Template Based**), extract one without any template (**Zero Shot**), find where each document starts in a multi-page PDF (**Split**), detect and straighten documents in a photo (**Crop**), and burn a redaction into a copy of a document (**Redaction**).

Every run is a real call to the Vision service. It is billed and rate-limited just like an API call, and every tool except Template Based extraction also fires your webhooks (see What every run costs and triggers).

How to get here	Vision left menu › **Playground**, or the **Playground** card on the Vision **Dashboard**
Who can use it	The shell grants the screen to the roles `cogneris-admin`, `cogneris-tech`, `customer-admin` and `customer-tech`. The service only checks that you are signed in. **Redaction** has its own conditions (see Redaction).
Content status	From source; not observed (`vision-fe` `0b724a0` + cogneris-vision-fe#192, `vision-be` `fce7fcc` + cogneris-vision-be#504, 2026-10-22)

The start screen

The screen is titled **Choose Extraction Mode**, with the subtitle *Upload a document to start extraction*. It shows two groups of tiles. Clicking a tile opens your computer's file picker. The tool starts as soon as you pick a file; there is no separate "run" button.

Group	Tile	Tile text	Accepted files	Section
Extraction mode	**Template Based**	*Classify the document first, then extract fields using a trained template* · **Click to upload**	Any image, or PDF	Template Based
Extraction mode	**Zero Shot**	*Auto-detect the document type and extract all key fields in a single pass* · **Click to upload**	Any image, or PDF	Zero Shot
Other tools	**Split**	*Identify each distinct document in a multi-page PDF and show where it begins* · **Click to upload a PDF**	PDF only	Split
Other tools	**Crop**	*Detect documents in an image and crop each detected region.* · **CLICK TO UPLOAD JPG, PNG OR PDF**	JPG, PNG, PDF	Crop
Other tools	**Redaction**	*Permanently hide sensitive content in a rasterized copy.* · see the table below	PDF, JPEG, PNG, WEBP	Redaction

Under the tiles: *Or drag and drop a file on either tile*. Dropping a file on **Template Based** or **Zero Shot** picks that mode and runs it, the same as clicking the tile and choosing the file. **Crop** also accepts a drop; **Split** does not.

The **Redaction** tile appears only when the service reports that redaction is available, or when it could not find out. Its last line changes with its state:

Tile text	Meaning	Clicking it
Open Redaction	At least one template has a published redaction policy you can use, or you hold `cogneris-admin` or `cogneris-platform-admin`	Opens the policy step
Requires a published Redaction policy	Redaction is on, but no template has a published policy yet	Nothing. The tile is disabled.
Could not check availability. Try again	The availability request got no answer	Asks the service again

Each Playground run handles **one file**. If you pick several, only the first is used.

Common problems on the start screen

Message	Cause	What to do
Unsupported file format. Please upload an image or PDF.	The file picked for **Template Based** or **Zero Shot** is not an image or a PDF	Convert the file to PDF, JPG or PNG
Split only accepts PDF files.	A non-PDF was picked for **Split**	Use a PDF
Crop only accepts JPG, PNG or PDF files.	Another format was picked for **Crop**	Use JPG, PNG or PDF
Unable to process the PDF. Please try again.	The **Split** call failed (any service error, including a rejected file or an exhausted quota)	Try again. If it keeps failing, check the file and your plan.
Unable to process the file. Please try again.	The **Crop** call failed	As above

Message	Cause	What to do
The model did not detect any documents.	**Crop** returned no document and no image	Use a sharper photo with the whole document in frame

The service enforces its own file checks on every tool: the content must match the extension, and the file can be at most 50 MB (Redaction: 100 MB, checked in the browser). A file that fails these checks shows the error message of the tool you used.

Template Based extraction

Two calls run in sequence:

1. **Classify.** The file is sent to the classifier, which returns a document type and a confidence. The service then looks for one of your templates whose name matches that type. It first tries an exact match (case-insensitive), then a normalized match.
2. **Extract.** If a template matched, the file is extracted against that template's schema. Because the template id is sent, the service builds the prompt from the template's fields and asks the model to return only the aliases the template declares (see Fields and schema).

Once the classifier answers and finds no template, the Playground stops. It does not extract with an empty schema; it shows the **No Template** state instead (see No template for this document type).

Template Based is the only mode that can show the **Document trust check** panel (see Document trust check).

Zero Shot extraction

A single call classifies the document and extracts its key fields at the same time, without a template. The result has the same layout as Template Based. The mode chip reads **Zero Shot**, and no **Document trust check** panel is shown: the service never runs the fraud side-car for zero-shot.

Use it to see what Vision reads on a document type you have not modeled yet. Then create a template and switch to Template Based to get stable field names.

The extraction result screen

Template Based and Zero Shot share one result screen. The document preview is on the left, the extracted data on the right.

While it runs

Element	What it shows
Status badge	**Identifying Document…** while classifying, then **Extracting Data…** (Template Based) or **Extracting (Zero Shot)…**, with animated dots and a running timer in seconds
Progress line	A message that changes every 5 seconds. Template Based: *still processing*, *identifying document type*, *getting document information*, *almost there*, *just a little more*. Zero Shot: *analyzing document structure*, *identifying key fields automatically*, *extracting data without template*, *almost there*, *just a little more*
Step list	**Uploaded** › **Classified document type** › **Extracting fields…** › **Trust & fraud check**, each marked done, current or pending
Data panel	Six placeholder rows

When it finishes

Element	What it shows
Status badge	**Document Processed Successfully** and the total time, **Process Error** on failure, or **No Template** when no template matched
New Upload (round upload button)	Clears everything and returns to the start screen
Type chip	The detected document type and its confidence, e.g. "Invoice · 92.50%". The confidence has two decimals and is colored: green from 90, amber from 50 to 89.99, red below 50, neutral when the service sent none.

Element	What it shows
Mode chip	**Template-based** or **Zero Shot**
Document trust check	Template Based only, when the fraud check ran (see below)
Form / JSON	Switches the data panel between field cards and raw JSON. Hidden while running, on error, and on **No Template**.

Preview toolbar

Button	What happens
+ (**Zoom In**)	Enlarges by 20 % per click
– (**Zoom Out**)	Shrinks by 20 % per click, down to about 40 %
⤢ (**Reset**)	Back to 100 % and no rotation
⟳ (**Rotate**)	Rotates 90° clockwise

PDFs are drawn page 1 only, on a canvas, at the zoom you choose. Images are scaled and rotated in place.

Form view: the extracted fields

The **Extracted Data** panel lists what the model returned:

Kind of value	How it is shown
Single value	A card with the field label, the value and, when the model returned one, the confidence as a whole number followed by `%` with a colored dot. The card's accent uses the same bands as the type chip: 90 and above high, 50 and above medium, below 50 low.

Kind of value	How it is shown
Group of values (an object)	Each inner key becomes a card of its own
List (an array, e.g. invoice lines)	A block titled with the field name, one row per item, each cell with its label and value. A cell that is itself a list becomes a nested table.

Click a card to select it. On an **image**, the area the model read the value from (its bounding box) is outlined on the preview. Click the card again to clear the outline. On a **PDF** no outline is drawn, even when the model returned a box.

The confidence is what the model reports about itself. It is not a measured accuracy. If nothing was extracted, the panel reads *No data extracted.*

JSON view

Shows the extracted values as indented JSON, with the confidence, label, page and box wrappers removed: you see the values only, in the shape your API integration will receive them.

No template for this document type

When the classifier names a type that none of your templates matches, the badge shows **No Template** and the panel shows **No Template for This Document Type** with one of two texts:

- *Document type detected as …, but no predefined template exists for this type. Use the Template creation module and then try again.*
- *This document couldn't be matched to any of your templates. Use the Template creation module to create one, then try again.* (when the type is empty or "UNKNOWN")

Create Template opens the **New Template** screen with the document you tested already attached and, when the classifier recognised a type, that type as the title.

Common problems with extraction

Message	Cause	What to do
Failed to run classification process. Please try again.	The classifier call failed (any error from the service, including a rejected file, a rate limit or an exhausted plan)	Try again later. If it persists, check your plan usage.
Unable to identify the document type.	The classifier answered with no result	Try a clearer scan, or use **Zero Shot**
Unable to extract data. The document was classified but extraction failed.	The extract call failed after a successful classification. This also covers the model returning truncated output, and a document the input-quality gate rejected as too poor to read.	Try again with a sharper, complete document
Unable to extract data. Zero-shot extraction failed.	The zero-shot call failed	Try again

Document trust check

An advisory panel that shows the fraud assessment of a Template Based extraction. It never blocks or hides the extracted data in the Playground.

When it appears

The service runs the fraud check (the `fraud-detection` agent) only when **all** of these hold. Otherwise no panel is shown at all:

1. Fraud-on-extract is enabled for the deployment.
2. The extraction was pinned to a template (Template Based, with a match).
3. Your tenant's agent guards allow the `fraud-detection` agent, agents are not paused, and fraud enforcement is set to **Advisory (record only)** (see Tenant agents).
4. The template's agent configuration enables the `fraud-detection` agent (see Agent).

The fraud check makes one extra model call, which estimates how likely the document is to be AI-generated. Its cost is added to the run.

What the panel shows

Element	What it shows
Document trust check	The panel title
Badge	**Trusted**, **Review recommended**, **Suspicious**, or **Verification unavailable**
Risk score	The combined score, 0–100 %
AI-generated likelihood	The model's estimate, 0–100 %, that the document was produced by AI
Policy action (advisory only)	`allow`, `flag` or `block`: what enforcement would have done. In the Playground the check only runs in advisory mode, so this always reads `allow`.
Analysis ID	The assessment id. Quote it to support, or find the run in the review queue.
Recommendation	Trusted: *Continue with the extracted data.* · Review: *Human review is recommended before relying on the result.* · Suspicious: *Pause the decision and perform human review; the extraction remains available.*
Coverage is incomplete. These checks could not complete:	Names of the checks that failed, plus `download:` followed by an id for any document the pipeline could not read. Only shown when something failed.
Evaluated checks	One row per check: its name in words with the technical name beside it, and **Clear**, **Flagged** or **Failed**, followed by the signals it raised
Signal	Severity (**Low**, **Medium**, **High**), the signal code, **Signal score** as a percentage, and the evidence text

When no assessment exists, or the request fails, the badge reads **Verification unavailable** and the panel says *The fraud assessment is unavailable. Extracted data remains available.* along with the **Analysis ID**. The panel never shows **Trusted** for an assessment it could not read.

For an assessment recorded before the checks were listed, the panel shows *Coverage is unknown because this assessment predates the recorded check roster.* and lists the signals without grouping them by check.

The check names, signal codes and evidence texts come from the service as they are, in English and not translated.

How the risk level is decided

The service combines three terms and clamps the result to 0–1:

- 0.6 × the **highest** score among the signals below
- 0.3 × the AI-generated likelihood
- 0.4 × the prompt-injection magnitude (0.7 when an injection phrase is found in the extracted text, otherwise 0)

At 0.35 or more the level is **Review recommended**; at 0.7 or more it is **Suspicious**; below 0.35 it is **Trusted**. These are the global defaults. Your tenant's fraud scoring settings and the template's sensitivity can change them. A broken or unverifiable PDF signature never yields **Trusted**: the level is raised to at least **Review recommended**, while the score itself is left unchanged.

The checks and their signals

The checks run in this order. A score of 0 % means the signal is informational and does not raise the risk.

Check	Signal code	Severity · score	What it means
`semantic`	`invalid_tax_id`	High · 90 %	A field whose name contains `cpf` or `cnpj` holds a number that fails the check-digit validation
`semantic`	`impossible_date`	Medium · 60 %	A date field (name contains

Check	Signal code	Severity · score	What it means
			`date`, `data`, `emiss` or `vencim`) holds a date in the future
`semantic`	`value_inconsistency`	Low · 40 %	An amount field (name contains `amount`, `total`, `valor` or `price`) holds something that is not a number
`pdf_structure`	`pdf_incremental_updates`	Medium · 50 %	The PDF has more than one end-of-file marker: it was edited and re-saved after creation
`pdf_structure`	`pdf_date_mismatch`	High · 80 %	The PDF's modification date is earlier than its creation date
`pdf_structure`	`pdf_producer_anomaly`	Medium · 50 %	The PDF's creator and producer differ (e.g. made in one tool, re-saved in another)

Check	Signal code	Severity · score	What it means
`pdf_structure`	`pdf_missing_metadata`	Low · 30 %	The PDF has neither creator nor producer metadata
`image_ela`	`ela_inconsistency`	Medium · 55 %	Error-level analysis: after a JPEG re-compression, more than 20 % of pixels changed noticeably, which suggests pasted or edited regions. At most one per document.
`perceptual_hash`	`duplicate_document`	High · 80 %	The exact same file (same SHA-256) was already submitted in your tenant
`perceptual_hash`	`layout_reuse`	Medium · 50 %	A visually near-identical document was already submitted in your tenant
`perceptual_hash`	`repeat_reuse`	Medium or High ·	The same visual fingerprint has 3 or more

Check	Signal code	Severity · score	What it means
		60 % up to 95 %	earlier submissions in your tenant. The score rises 10 % per extra submission, and the signal is High from 70 %.
`c2pa_provenance`	`c2pa_present_invalid`	High · 70 %	The file carries a content-provenance (C2PA) manifest that is broken
`c2pa_provenance`	`c2pa_present_valid`, `c2pa_present_untrusted`, `c2pa_absent`	Low · 0 %	A valid manifest, a manifest from an issuer you have not trusted, or no manifest. Informational only.
`c2pa_provenance`	`c2pa_unavailable`	Low · 0 %	Provenance was not evaluated for this document
`pdf_signature`	`pdf_signature_broken`	High · 70 %	The PDF's digital signature no longer matches its content. Forces at least **Review recommended**.

Check	Signal code	Severity · score	What it means
`pdf_signature`	`pdf_signature_unverifiable`	High · 70 %	The signature could not be checked. Forces at least **Review recommended**.
`pdf_signature`	`pdf_signature_untrusted`	Medium · 40 %	Signed, but by a certificate that is not trusted
`pdf_signature`	`pdf_signature_valid`, `pdf_signature_absent`	Low · 0 %	Validly signed, or not signed. Informational; a valid signature does not lower the risk.
`pdf_signature`	`pdf_signature_unavailable`	Low · 0 %	The signature was not evaluated for this document
`prompt_injection`	`prompt_injection_attempt`	High · 0 %	The extracted text contains an instruction aimed at the AI (e.g. "ignore previous instructions", "system prompt", "approve this"). It is listed at 0 %, but it raises the risk through the prompt-

Check	Signal code	Severity · score	What it means
			injection term above.
`history_velocity`	`velocity_anomaly`	Medium · 60 %	Your tenant submitted more than 20 documents for fraud assessment so far today (default window and limit)

What to do with the result: on **Review recommended** or **Suspicious**, have a person look at the original document before relying on the data. Confirmed cases and false positives are recorded in the review queue (see Tenant agents).

Split

For a multi-page PDF that contains several documents (an ID, a proof of address, a payslip…). The service identifies each distinct document and the page where it begins. It does not cut the PDF into files.

Input: one PDF. The browser rejects anything else, and the service rejects it again by extension and content.

While it runs: Processing document… with the file name, a timer, and a step that advances every 2.5 seconds: *Uploading document…*, *Analyzing pages with AI…*, *Detecting document types…*, *Building the document index…*, *Opening the original PDF…*. These steps are timed in the browser and do not report the service's real progress. **Cancel** returns to the start screen; the request already sent to the service is not withdrawn.

Result:

Element	What it does
Document index and file name	Header
New Upload	Back to the start screen
Document index table: **Document type**, **Start page**	One row per document found. Click a row to jump to its first page. The row of the document you are viewing is highlighted (the last document whose start page is at or before the current page).
No distinct documents were identified. You can still review the original PDF.	The service found nothing usable. If any entry has no type or an invalid start page, the whole index is discarded.
‹ / **›** and the page box, **of …**	Previous and next page, or type a page number and press Enter
– / **+** and the percentage	Zoom from 50 % to 200 % in steps of 25 %
The original PDF could not be displayed. Try a different file.	The browser could not render the PDF

Crop

Detects the documents in a photo or scan, corrects their perspective, and shows the straightened image. This is useful for a phone photo of an ID card on a table.

Input: JPG, PNG or PDF. A PDF of **exactly two pages** (typically front and back) is merged in the browser into one PNG image, page 1 above page 2, before it is sent. Any other PDF is sent as it is.

While it runs: Processing document…, the file name, a timer and timed steps: *Uploading file…*, *Detecting documents with AI…*, *Computing bounding boxes…*, *Cropping detected regions…*, *Preparing crop result…* **Cancel** works as in Split.

Result: two panels.

Panel	What it shows
Original	The file you sent. On an image (including a merged two-page PDF), each detected document is outlined. A PDF that was not merged is shown without outlines.
Image Cropped	The composite the service returns when there is one; otherwise every detected document's image, one under the other. If there is none: *Select a row to preview the cropped image.*
New Upload	Back to the start screen

When several documents are detected, only one image is shown here. The detection table (type, coordinates, angle, confidence) that the translation catalog still describes is not shown in this version.

Redaction

Produces a **rasterized copy** of a document in which sensitive regions are painted over in the pixels, so they cannot be recovered by copying text. The categories, style and retention come from a **redaction policy** published on a template (see Redaction policy).

When you can use it

- The service must report redaction as available for your tenant.
- At least one template must have a published policy. Otherwise the tile reads **Requires a published Redaction policy**.
- Platform administrators (`cogneris-admin` , `cogneris-platform-admin`) can open the tool regardless, but can only submit a file when a policy is available.

Step 1 — Redaction policy

Field	What it does	Required	Values	Validation messages
Policy template	The template whose	Yes	Placeholder **Choose a template with a published policy**,	*Unable to load policies · No template has a*

Field	What it does	Required	Values	Validation messages
	published policy is applied		then every template with a published policy, by name. The list loads all pages automatically, up to 20 pages.	*published redaction policy yet. Publish one from a template's Redaction tab.* · *Redaction is unavailable for this template.*

Button	What happens
Back	Back to the start screen
Select document	Opens the file picker; enabled once a valid policy is selected. The upload starts as soon as you pick a file.

File checks in the browser: PDF, JPEG, PNG or WEBP, at most 100 MB. The service processes at most 100 pages.

Step 2 — Redacting document

Redaction runs as a background job. The screen polls its status (every 1 s at first, slowing to every 5 s) and shows the current stage: *Queued*, *Rasterizing pages*, *Detecting sensitive regions*, *Consolidating regions*, *Burning redactions into pixels*, *Verifying redacted output*, *Storing temporary artifacts*. It also shows pages done out of the total, once known.

Cancel asks the service to stop the job. If the service does not confirm: *The cancellation could not be confirmed. Please try again.*

Step 3 — the result

Before anything is shown, the browser checks both files against the job's manifest (fingerprints, page counts, categories) while showing *Verifying downloaded artifacts*. If anything does not match, the result is withheld: *The result could not be verified and is unavailable.*

Element	What it shows or does
Redacted document ready	Header, with **New Upload**
Original / Redacted result	Side by side, on the same page and zoom. For PDFs: **‹ / ›**, **Page … of …**, zoom 50–200 %.
Safe redaction summary	**Protected categories** (each category with its count), **Affected pages** (or **None**), **Low-confidence regions were redacted and require review** (with the count, only when there are any), **Policy version**, **Duration** (… seconds), **Available until** (local date and time), **Output hash** (the first 12 characters)
Download redacted document	Saves `redacted-` followed by the first 8 characters of the job id, as `.pdf`, `.jpg` or `.png`
Download safe manifest	Saves `redaction-manifest-` followed by the same 8 characters, as `.json`. The manifest lists regions, categories and fingerprints, not the hidden content.

Possible categories: **Full name**, **Tax ID**, **Document number**, **Date of birth**, **Address**, **Phone**, **Email**, **Bank account**, **Payment card**, **Face**, **Signature**, **QR or barcode**, **Passport**.

The result is temporary. When **Available until** passes, the screen drops it and shows *The temporary result has expired.*

Copy to ECM

Shown only when the service allows copying for your tenant.

Field	What it does	Required	Format	Validation
Destination folder ID	The ECM folder that receives the copy	Yes	A folder UUID (not the all-zero one)	**Copy to ECM** stays disabled until it is valid
Document name	Name of the new ECM document	Yes	1 to 200 characters, no control characters	As above
I confirm that this redacted artifact should be copied to ECM.	Explicit confirmation	Yes	Checkbox	As above

Copy to ECM (shows **Copying to ECM** while it runs) copies the redacted file only, never the original. On success: *Copied as document …, version …*., and the form locks. The copy is accepted only if the service returns valid ids and a fingerprint equal to the verified output.

Common problems with redaction

Message	Cause	What to do
Redaction only accepts PDF, JPEG, PNG or WEBP files.	Wrong file type	Convert the file
This file is larger than the 100 MB maximum.	File too large	Split or compress it
Redaction is unavailable for this template.	The policy could not be read, or the feature is off	Pick another template, or ask an administrator

Message	Cause	What to do
Sensitive data was still visible after redaction, so the result was withheld. Widen the categories on this template and run it again.	The final check still found sensitive data	Add categories to the policy and run again
The redacted document did not pass verification, so it was not released.	Output verification failed	Run again; report it if it repeats
Sensitive-data detection could not finish on this document.	Detector failure	Run again later
The redacted document could not be stored securely.	Storage failure	Run again later
The pages of this document could not be rendered for redaction.	The pages could not be rasterized	Try another export of the file
This file could not be read as a redactable document.	Corrupt or unsupported content	Check the file
The redaction policy on this template is not valid. Ask an administrator to review it.	Invalid policy	Fix the policy (see Redaction policy)
This redaction stopped because the job was changed somewhere else.	The job was changed elsewhere	Start again
Redaction is temporarily unavailable. Please try again in a few minutes.	Any other service-side failure	Wait and retry
This redaction expired before it finished.	The job expired	Start again
Failed	A request failed without a specific reason	Retry

Message	Cause	What to do
The document preview could not be rendered.	The browser could not draw the files	The downloads may still work. Otherwise run again.
The authenticated download could not be completed.	Download failed or returned an unexpected file	Retry
The ECM copy could not be completed safely.	The copy failed or returned an unexpected answer	Check the folder ID and your ECM access, then retry

On the failure card, the title shows the final state (**Completed**, **Failed**, **Cancelled** or **Expired**), the reason, and **New Upload**.

The old Chat address

`/playground/chat` used to hold an in-app chat. It now redirects to the **MCP** screen, which connects an editor to the Cogneris MCP server (see MCP). Old bookmarks and links land there.

What every run costs and triggers

Topic	What happens
Billing	Classify, Extract, Zero Shot, Split and Crop are billable calls, charged as usage of your plan. Template Based makes **two** billable calls (classify, then extract), plus the AI-likelihood model call when the fraud check runs. On a plan that stops at its limit, the service refuses the call once the monthly allowance is used, and the Playground shows that tool's generic failure message. The Playground does not display a cost.
Rate limit	These calls share the service's `document-processing` quota. When you exceed it, calls are refused and the tool shows its failure message.
Webhooks	The Playground asks the service not to publish webhooks on every call — classify, extract, zero-shot, split and crop — so a try never fires

Topic	What happens
	`*.processed` or `*.failed` to your configured endpoints (see Developers).
What is stored	The fraud assessment and the fingerprint of the document (used later for duplicate detection) are stored when the fraud check runs. A redaction job's files are kept until **Available until**.

Worked example: create a template, test it, read the result

1. **Create the template.** In **Templates**, create a template. Give it a **Title** that names the document type the classifier will return, for example `Invoice`: the Playground finds your template by comparing its name with the detected type. Upload 2 to 5 sample invoices and wait for processing to finish (see Templates).
2. **Shape the fields.** In the template editor, keep the fields you need, with the aliases your integration expects (see Fields and schema).
3. **Test it.** Open **Playground**, click **Template Based**, and pick an invoice that was **not** among the samples.
4. **Read the chips. Invoice · 96.00%** in green with **Template-based** means the document was classified and matched to your template. If you see **No Template**, the detected type differs from your template's title. Note the type shown in the message and rename the template to match, or create one with that name.
5. **Read the fields.** Each card shows a value and the model's confidence. Click a card to see where it was read (on images). Treat amber and red cards (below 90) as candidates for clearer field descriptions in the schema, or for a better sample.
6. **Check the JSON.** Switch to **JSON**: these are the keys and values your API call will return.
7. **Read the trust check** if it appears. **Trusted** means continue. **Review recommended** or **Suspicious** lists the signals that raised the score; for example, `pdf_incremental_updates` means the PDF was edited after creation.
8. Change the schema, run the same file again, and compare.

Related

- Vision overview
- Templates · Fields and schema · Redaction policy · Agent

- Tenant agents: agent guards, fraud enforcement and scoring, fraud review queue
- Developers: API keys, webhooks and request logs
- MCP
- Workflow phases that run the same tools: Extraction, Classification, Zero Shot, Split, Crops

---

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Tenant agent settings

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Company-wide rules for the AI agent engine that runs your templates: a monthly spending cap, which providers, regions and models the agents may use, what is stored about each call, how fraud findings are enforced and an emergency stop. The same screen shows this month's agent spending and the queue of documents that fraud detection flagged for a human verdict.

How to get here	Vision menu › **Tenant agent settings**. The screen has three tabs: **Guards**, **Analytics** and **Review**. Opening the screen lands on **Guards**.
Who can use it	Every signed-in member of your company can open all three tabs. Only a user with the `tenant.admin` role can save **Guards** and submit a fraud verdict on **Review**; everyone else sees the same fields, disabled. A Cogneris platform administrator can additionally set a cost-cap override for your company (see Platform administrator override).
Content status	From source; not observed (`vision-fe` `0b724a0` , `vision-be` `fce7fcc` , 2026-10-22)

The screen title is **Tenant agent settings**, with the line *Configure tenant-wide guardrails for the AI agent engine and review month-to-date analytics.*

Guards

One form with every guard of your company. The settings apply to every template's agent runs; a template's own agent settings (see Template agent) work inside these limits.

When your account is not a tenant administrator, the tab opens with the notice *Only tenant admins can edit these settings. The form below is read-only for your account.* Every field is

shown but disabled, and the **Save** button is not shown.

Fields

A list field takes values separated by commas; spaces around each value are ignored. Under each list field the screen says *Comma-separated. Leave empty to allow everything.*

Field	What it does	Required	Format / values	Default	Validation messages
Monthly cost cap (USD)	Your company's spending ceiling for agent runs in the current calendar month (UTC). Before each run that is not a dry run, the service adds up this month's agent cost so far; when it has reached the cap, the run is refused with a monthly-cap error before any model is called. `0` means no cap. On save, the service checks the amount against the	Yes	Number, 0 or more, cents allowed. Placeholder `$0.00`.	`0`	*Cap must be 0 or higher.* (shown once you leave the field)

Field	What it does	Required	Format / values	Default	Validation messages
	maximum your subscription plan allows (see Saving the guards).				
Active override (set by platform admin)	Not a field: a line under the cap, followed by an amount, shown only while a Cogneris platform administrator has set a cost-cap override for your company. See Platform administrator override.	—	Amount in USD	Not shown	—
Allowed providers	Restricts agent runs to models from these AI providers.	No	Provider names, for example `google-vertex`, `openai`, `anthropic`. Compared without regard to case; spaces and	Empty (every provider)	—

Field	What it does	Required	Format / values	Default	Validation messages
			underscores count as hyphens, and a trailing `-ai` is ignored.		
Allowed regions	Restricts agent runs to models served from these regions.	No	Region names, for example `us-central1`, `global`. Compared like providers.	Empty (every region)	—
Allowed model classes	Restricts agent runs to models of these classes.	No	Class names, for example `general`, `high_stakes`. Compared like providers.	Empty (every class)	—
Allow fraud detection in Playground	Adds the fraud-detection agent to your company's allowed domain agents. The Playground offers a fraud check only when this box is ticked, agents are not paused,	No	Checkbox	Off	—

Field	What it does	Required	Format / values	Default	Validation messages
	Fraud enforcement is **Advisory (record only)**, the template has fraud detection enabled, and Cogneris has the feature turned on for the platform. Under the box: *Every template must also opt in to fraud detection separately.*				
Allowed domain agents	The domain agents (specialist agents such as fraud detection) that a template may enable. A run whose template enables an agent that is not on the list is refused	No	Agent keys, for example `fraud-detection`, `gl-coding`. Exact match.	Empty (every agent)	—

Field	What it does	Required	Format / values	Default	Validation messages
	once the list names any agent other than `fraud-detection`; a list that holds only `fraud-detection` (the checkbox above) still allows every agent. Agents that only a workflow step runs are not checked against this list. Typing `fraud-detection` here is moved to the checkbox above; duplicates are dropped.				
Prompt storage mode	What the service keeps of the prompt sent to the model at each agent step. A hash	Yes	**Hashed (default — prompt content stored as a hash for audit only)** —	Hashed	—

Field	What it does	Required	Format / values	Default	Validation messages
	of the prompt is always kept.		hash only. **Prompt only (the prompt is stored, the model output is not)** — the prompt text is stored too. **Full (prompt + completion stored — useful for evaluation)** — the prompt text is stored too.		
Response storage mode	What the service keeps of the model's answer at each agent step. A hash of the answer is always kept. Uses the same three options as **Prompt storage mode**: only **Full** stores the answer text; **Hashed** and **Prompt**	Yes	Same three options	Hashed	—

Field	What it does	Required	Format / values	Default	Validation messages
	only keep the hash alone.				
Require no-training terms	Every model a run could use must be marked, in the Cogneris model catalog, as covered by terms that exclude training on your data.	No	Checkbox	Off	—
Require zero-retention processing	Every model a run could use must be marked, in the Cogneris model catalog, as processing with zero data retention.	No	Checkbox	Off	—
Minimum field confidence	After a run, each extracted field whose independent validator confidence is below this	Yes	Number from 0 to 1, two decimals. Placeholder `0.85`.	`0.85`	*Enter a confidence value from 0 to 1.* (shown once you leave the field)

Field	What it does	Required	Format / values	Default	Validation messages
	value is listed as a low-confidence field and the run is marked as needing human review. A run is also marked for review when it enables a domain agent whose key contains `fraud`, `risk` or `adverse`, or when your AI policy routed it to review.				
Fraud enforcement	What a fraud-detection risk level does to a run. **Advisory (record only)**: the assessment is recorded and nothing is blocked. **Enforce**: a risk level at or above	Yes	**Advisory (record only)**, **Enforce**	Advisory	—

Field	What it does	Required	Format / values	Default	Validation messages
	Block at risk level is recorded as a block; a lower flagged level is recorded as flagged. Whether a block actually withholds the extraction result also depends on a platform-wide setting that Cogneris controls; when that setting is off, a block is recorded only. In **Enforce** the Playground does not run a fraud check.				
Block at risk level	The lowest risk level that counts as a block under **Enforce**.	Yes	**Review or higher**, **Suspect only**	Suspect only	—

Field	What it does	Required	Format / values	Default	Validation messages
	Ignored under **Advisory**.				
Fraud scoring (advanced)	A group of five overrides for how fraud risk is scored. The group says *Each setting comes from the template's fraud sensitivity preset unless you enable an override. Weights ≥ 0; thresholds between 0 and 1; review ≤ suspect.* Each row is a checkbox; ticking it shows a number box for your value. Unticked, the row is saved empty and the default applies. See	No	See the next table	All unticked	**Review threshold** ≤ **Suspect threshold** (when both thresholds are overridden and review is higher). A negative weight or a threshold outside 0–1 disables **Save** without a message.

Field	What it does	Required	Format / values	Default	Validation messages
	Fraud scoring.				
Pause all agent runs (kill switch)	Stops the agent engine for your company: every agent run is refused with an agents-paused error, and the Playground fraud check is not offered. Under the box: *When enabled, the engine refuses to run agents for this tenant until you turn it back off.*	No	Checkbox	Off	—

Fraud scoring

Override	What it changes	Default when not overridden
Deterministic signal weight	Weight of the rule-based tamper signals in the risk score.	The platform default; the placeholder shows `0.6`.
AI-generated likelihood weight	Weight of the likelihood that the document was	The platform default; the placeholder shows `0.3`.

Override	What it changes	Default when not overridden
	generated by AI.	
Prompt-injection weight	Weight of the likelihood that the document carries instructions aimed at the AI.	The platform default; the placeholder shows `0.4`.
Review threshold	Score from which a document is rated *review*.	The fraud sensitivity of the template: `0.50` low, `0.35` medium, `0.20` high; medium when none is set. The placeholder shows `0.35`.
Suspect threshold	Score from which a document is rated *suspect*.	The fraud sensitivity of the template: `0.85` low, `0.70` medium, `0.55` high; medium when none is set. The placeholder shows `0.7`.

The service clamps what it uses: a weight never goes below 0 and a threshold stays between 0 and 1.

How the model restrictions are applied

When any of **Allowed providers**, **Allowed regions**, **Allowed model classes**, **Require no-training terms** or **Require zero-retention processing** is set, the service checks, before a run starts, every model the run could use — the main model and the fallback model of each step, and any model a template pins to a step. Each one must be active in the Cogneris model catalog and satisfy every restriction. If one does not, the whole run is refused with a model-not-allowed error, before any document is sent to a model. If none of the run's models can be determined, the run is refused too.

Buttons and actions

Button	What happens	Who	Reversible?
Save (**Saving…** while it runs)	Sends the whole form. On success: *Guards saved.* Disabled while a field is invalid, while saving, and when your subscription plan could not be loaded. Hidden for users who are not tenant administrators.	`tenant.admin`	Yes — change the values and save again.
Retry	Shown with the load error. Loads the guards again.	Everyone	—

Saving the guards

Saving needs your company's subscription plan, which the screen loads with the guards. The service then compares **Monthly cost cap (USD)** with the plan's maximum monthly cap: above it, the save is refused (the service's message asks you to upgrade your plan). A plan maximum of `0` means unlimited. This check is skipped while a platform administrator override is active.

Before your company saves the guards for the first time, the service applies the defaults listed in the Fields table (no cap, nothing restricted, hashed storage, `0.85` confidence, advisory enforcement).

Platform administrator override

A Cogneris platform administrator — not a tenant administrator — can set two things for your company outside this screen:

- **A cost-cap override.** While it is set, the screen shows *Active override (set by platform admin)* with the amount, **Analytics** measures spending against that amount, and saving your guards no longer checks your cap against the plan maximum. The monthly limit that refuses runs is still your **Monthly cost cap (USD)**. Clearing the override restores the plan check.
- **A protected-health-information scope.** When it is on, every model a run could use must be eligible under a business associate agreement, and **Require no-training terms** and **Require zero-retention processing** apply as if ticked, whatever the form says. Your administrators cannot set or clear this scope.

Common problems

Message	Cause	What to do
Couldn't load tenant guards. (followed by the HTTP status when there is one)	The guards or your subscription plan could not be read.	Use **Retry**. A `401` means your session expired; sign in again.
Couldn't save tenant guards. (followed by the HTTP status)	The service refused the values — for example a cap above the plan maximum, or the plan's maximum could not be read — or you are not a tenant administrator (`403`).	Check the cap against your plan, then save again.
Plan unavailable; refresh and try again.	The subscription plan was not loaded. In practice **Save** is simply disabled in this state.	Reload the page.
Save stays disabled with no message	A fraud-scoring override is negative or a threshold is outside 0–1, or the plan was not loaded.	Correct the value or reload the page.
Every agent run fails	**Pause all agent runs (kill switch)** is ticked, the monthly cap is reached, or a model restriction excludes a model the template uses.	Untick the kill switch, raise the cap, or widen the restriction.

Analytics

Your company's agent spending and run counts for the current month, from the first day of the month to today.

The numbers come from a daily summary of agent runs. When that summary cannot be read, the service shows the last copy it kept, with the banner *Showing cached data — analytics export was unavailable.*

Fields

Field	What it shows	Format
Last updated	The most recent day included in the figures.	Date (`yyyy-MM-dd`), or `–` when there is no data.
% of monthly cap consumed	This month's total agent cost divided by your monthly cap — the platform administrator override when one is active, otherwise **Monthly cost cap (USD)**. Below it, the amount spent and the cap (for example `$12.40 / $100.00`).	Percentage; `–` when no cap is set.
Total runs	Number of agent runs this month.	Whole number
Success rate	Successful runs divided by all runs.	Percentage; `–` when there are no runs.
Per-template table: **Template**	The template, shown by its ID.	Template ID
Runs	Runs of that template this month.	Whole number
Success rate	Successful runs of that template divided by its runs.	Percentage
Cost	Agent cost of that template this month. The table is sorted by cost, highest first.	USD

When the month has no runs: *No runs recorded yet this month.*

Common problems

Message	Cause	What to do
Failed to load analytics (or the service's own error text)	The run summary or the guards could not be read.	Reload the page later.
No runs recorded yet this month. although agents ran	The run summary could not be read and the service had no saved copy; the screen shows this instead of an error.	Reload the page later.

Review

The **Fraud review queue**: documents that fraud detection rated *review* or *suspect* and that nobody has given a verdict yet, newest first, up to 100. A verdict is the human check on the automated assessment.

Above the queue, a strip summarises the verdicts of the last 30 days, one entry per risk level and verdict, for example `suspect / confirmed_fraud: 3`. The risk levels and verdicts appear as the service names them, untranslated.

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Risk	The risk level fraud detection gave the document.	—	`review` or `suspect`	—	—
Score	The risk score.	—	Number, two decimals	—	—
Flagged	When the assessment was made.	—	Date and time	—	—

Field	What it does	Required	Format / values	Default	Validation messages
Assessment panel	Opens with **Review**: the full fraud assessment of the document — risk badge, score, AI-generated likelihood, signals, evaluated checks and recommendation, as described in Playground.	—	—	—	—
Verdict	Your conclusion.	Yes, to submit	**Confirmed fraud**, **False positive**, **Inconclusive**	None selected	—
Notes	Free text kept with the verdict. Placeholder *Notes (optional)*.	No	Text	Empty	—

Buttons and actions

Button	What happens	Who	Reversible?
Review	Opens or closes the assessment and the verdict controls under the row.	Everyone	—

Button	What happens	Who	Reversible?
Submit verdict	Records the verdict and notes with your email as reviewer. The row leaves the queue and the 30-day strip is refreshed. Disabled until a verdict is chosen; for non-administrators it is always disabled and its tooltip says *Only a tenant admin can submit a verdict*.	`tenant.admin`	Not from this screen: once submitted, the document no longer appears in the queue.

Common problems

Message	Cause	What to do
No flagged runs to review.	No assessment at *review* or *suspect* is waiting for a verdict — or the queue could not be loaded, which the screen does not distinguish.	Reload the page to be sure.
The service's error text next to **Submit verdict**	The verdict was refused — for example the assessment no longer exists, or your account is not a tenant administrator.	Reload the queue.

Related

- Template agent — per-template agent settings, including fraud detection and its sensitivity.
- Playground — run a template and see the fraud assessment panel.
- Vision administration overview

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Developers

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

The Developers area holds what your own systems need to call the Cogneris API and to be called back by it: API keys, webhook endpoints, and the logs of API calls and webhook deliveries. The API itself is described in the developer portal.

How to get here	Vision menu › **Developers** › **API Keys**, **Webhook Endpoints**, **API Request Logs** or **Webhook Request Logs**. Opening **Developers** itself lands on **API Keys**.
Who can use it	Tenant administrators: the `customer-admin` and `customer-tech` profiles, and Cogneris operators (`cogneris-admin` , `cogneris-tech`) inside their own company. The services refuse every other member. When a list cannot be loaded — including when the service refuses a member without access — the screen says so instead of showing an empty list.
Content status	From source; not observed (`vision-fe` `0b724a0` + cogneris-vision-fe#192, `vision-be` `fce7fcc` + cogneris-vision-be#504, `platform-be` `166c01e` , `billing-be` `12ce77c` , 2026-10-22)

API Keys

API Keys — *Manage keys used to authenticate against the Cogneris API.* An API key is a secret your server sends with each API call. Each key belongs to one **application** and one **environment**:

- **Sandbox** keys start with `xtkt_test_` and reach only sandbox data.
- **Production** keys start with `xtkt_live_` and act on real data.

After the prefix, a key has 43 random letters and digits. Cogneris stores only a hash of the key; the full key is shown once, when it is created.

The list does not include developer tokens for the MCP server; those are managed on the MCP page.

Fields

The list:

Field	What it shows	Format
Name	The name given at creation.	Text
Key	The key, masked: its prefix, eight dots and its last four characters.	For example `xtkt_live_••••••••a1B2`
Application	The application the key belongs to.	Text, or `-`
Environment	**Sandbox** or **Production**.	Badge
Access	The key's scopes, or **Full access** (highlighted) for a key that can do everything — including every key created before access could be chosen.	Comma-separated scope names, or badge
Expires	The date the key stops working, or **Never** (highlighted) for a key with no expiry.	Date in your language's format, or badge
Created	Creation date.	Date in your language's format
Last used	Date the key was last used, or **Never**.	Date in your language's format
Status	**Active** or **Revoked**. Revoked keys stay in the list, greyed out.	Badge

Field	What it shows	Format
Actions	**Revoke**, on active keys only.	—

The **New API Key** dialog:

Field	What it does	Required	Format / values	Default	Validat messa
Name	A label for the key. Hint: *A short description so you can identify the key later.*	Yes	Up to 120 characters. Placeholder *e.g. Production backend*.	Empty	**Genera key** sta disable while empty.
Application	The application the key is minted in.	Yes, when shown	Your company's applications	The first application	—
Environment	Where the key works. Hint: *Sandbox keys start with xtkt_test_ and only reach sandbox data. Production keys start with xtkt_live_*	Yes, when shown	**Sandbox**, **Production**	**Sandbox**	—

Field	What it does	Required	Format / values	Default	Validat messa
	and touch real data.				
Access	What the key can do. Hint: *Pick the smallest access your integration needs. Full access is available, but never the default.* Each choice explains itself below the field.	Yes	See the table below	None — placeholder *Choose what this key can do*	**Genera key** sta disable until an access chosen.
Scopes	Shown only for **Custom**: one checkbox per scope. Hint: *Tick each scope the key needs.*	Yes, for **Custom**	Scope names, for example `workflows.read`	None ticked	**Genera key** sta disable while nc scope is ticked.
Expires after	How long the key	Yes	**30**, **90**, **180** or **365 days**	**90 days**	—

Field	What it does	Required	Format / values	Default	Validat messa
	works. Hint: *Every key expires. Create a new key before this one stops working.*				

The **Access** choices:

Choice	Scopes the key gets
Extraction	`classifiers.read`, `classifiers.write`, `playground.run` — classifiers and the playground.
Documents	`ecm.documents.read`, `ecm.documents.write`, `ecm.folders.read`, `ecm.folders.write`, `ecm.search.read` — the Document Center.
Portal	`portal.forms.read`, `portal.magiclinks.write` — read forms and send magic links.
Workflow	`workflows.read`, `workflows.write` — read workflows and runs, publish and start runs.
Read only	`classifiers.read`, `ecm.documents.read`, `ecm.folders.read`, `ecm.search.read`, `portal.forms.read`, `workflows.read` — nothing is changed.
Custom	The scopes you tick. Besides the ones above, `webhooks.manage` (webhook endpoints), `apikeys.manage` (create and revoke keys through the API, only with scopes the key itself has) and `builder.run` (Build AI, which spends credits).

Choice	Scopes the key gets
Full access	`*` — everything the API allows, including managing keys and Build AI. Use it only when no narrower choice fits.

When the applications cannot be loaded, **Application** and **Environment** are not shown and the dialog says *Applications could not be loaded; the key will be created in the default production environment.*

A key created on this screen always has the access you chose and always expires; after the expiry date the service refuses it. Revoking stops it earlier. Developer-token scopes (`mcp.read`, `mcp.write`) are not offered here — those keys are created on the MCP page.

Keys created before this choice existed have **Full access** and **Never** expire. While any of them is still active, the list says *… active key(s) have full access or never expire. Create a replacement with limited access and an expiry, then revoke the old key.* Nothing is revoked for you: create the replacement, move your integration to it, then **Revoke** the old key.

Buttons and actions

Button	What happens	Who	Reversible?
New API Key	Opens the **New API Key** dialog.	Tenant administrators	—
Generate key (**Generating…** while it runs)	Creates the key and shows it once, with *Save this key now — you will not see it again.*, the environment badge and the application name.	Tenant administrators	Revoke the key to undo.
Copy / **Copied!**	Copies the new key to the clipboard.	—	—
Done	Closes the dialog. The full key cannot be shown again.	—	—

Button	What happens	Who	Reversible?
Cancel	Closes the dialog without creating a key.	—	—
Revoke	Asks **Revoke API key** — *After revoking, … can no longer be used to authenticate. This operation is irreversible.* — then **Yes, revoke** revokes the key and records who revoked it.	Tenant administrators	No

Common problems

Message	Cause	What to do
No API keys yet. Create your first to start using the API.	No integration key exists — or the list could not be loaded (for example your profile is not a tenant administrator), which the screen does not distinguish.	Check your profile with an administrator.
Applications could not be loaded; the key will be created in the default production environment.	The applications service did not answer.	Close the dialog and try later if you need a sandbox key.
The dialog stays open after **Generate key** with the service's reason, or *The key could not be created. Please try again.*	The service refused the request or did not answer.	Read the reason; try again; check that you are a tenant administrator.
The key could not be revoked. Please try again. (or the service's reason)	Revoking failed; the key is still active.	Try again.

Message	Cause	What to do
Couldn't load your API keys. Please try again.	The list could not be read.	Reload the page.
Loading keys…	The list is loading.	—
… active key(s) have full access or never expire. …	At least one active key was created with full access or without an expiry.	Create a replacement with the access your integration needs, switch to it, then revoke the old key.

Webhook Endpoints

Webhook Endpoints — *Send HTTPS callbacks to your own servers when operations complete*. An endpoint is a URL on your server that Cogneris calls with a JSON message when one of the chosen events happens.

Endpoints created on this screen receive the events of your **production** environment. Sandbox endpoints are created through the API with a sandbox key and do not appear here.

Fields

The list:

Field	What it shows	Format
Name	The endpoint name.	Text
URL	The endpoint URL.	Text
Events	The subscribed events: the first two, then `+` and the count of the others.	For example `extract.processed, extract.failed +2`
Status	**Active** or **Inactive**. Inactive endpoints are greyed out and receive nothing.	Badge

Field	What it shows	Format
Created	Creation date and time.	Date and time
Actions	**Edit**, **Delete**.	—

The **Create webhook endpoint** / **Edit webhook endpoint** dialog opens with *All endpoints must be HTTPS.*

Field	What it does	Required	Format / values
Endpoint name	A label for the endpoint.	Yes	Up to 100 characters. Placeholder *e.g Service X webhook*.
Endpoint URL	Where Cogneris sends the events. The service also refuses URLs that point to private, loopback or link-local addresses, and checks the address again at every delivery.	Yes	Absolute `https://` URL, up to 500 characters. Placeholder *https://hook.yourdomain.com/exampl*
Select events to subscribe to	The events this endpoint receives. Hint: *Choose at least one event.*	Yes	See Events.
Advanced options › **Custom request headers (optional)**	Headers added to every request to this endpoint, for example your own authentication header. Hint: *Sent with every request to this endpoint. One*	No	**Header name**, **Value** pairs

Field	What it does	Required	Format / values
	row per header. Rows with an empty **Header name** are dropped. Stored encrypted.		
Advanced options › Override request body (optional)	Replaces the event message with this text. Hint: *If set, replaces the default event payload sent to the endpoint. Keep blank to receive the default JSON.* The text is sent as is, as `application/json`, and the signature is computed over it. Stored encrypted.	No	Text
Active	Shown only when editing. Unticked, the endpoint stops receiving events but is kept.	—	Checkbox

Advanced options opens by itself when the endpoint being edited already has headers or a body.

Events

Event	Sent when
`extract.processed`, `extract.failed`	An extraction finishes or fails.

Event	Sent when
`zeroshot.processed`, `zeroshot.failed`	A zero-shot extraction finishes or fails.
`crop.processed`, `crop.failed`	A crop operation finishes or fails.
`split.processed`, `split.failed`	A document split finishes or fails.
`classifier.processed`, `classifier.failed`	A classification finishes or fails.
`redaction.processed`, `redaction.failed`	A redaction job finishes or fails.
`facematch.processed`, `facematch.failed`	A face-match job finishes or fails.

Signing secret

When an endpoint is created, the dialog shows its **Signing secret** once, with *Save this signing secret now — you will not see it again.* The secret starts with `whsec_` followed by 64 hexadecimal characters.

Changing the **Endpoint URL** of an existing endpoint replaces the secret: after **Save endpoint** the dialog shows the new secret, once. Changing anything else keeps the secret. This screen has no other way to see or replace a secret.

How each request is signed

Every delivery is an HTTPS `POST` with these headers:

Header	Value
`X-Xtrakt-Signature`	The signature, in the format below.
`X-Xtrakt-Event`	The event name, for example `extract.processed`.
`X-Xtrakt-Delivery`	An ID that stays the same across every attempt of the same event at the same endpoint. Use it to ignore duplicates.

Header	Value
`X-Xtrakt-Delivery-Attempt`	The attempt number, starting at `1`.
Your custom headers	As configured.

The signature header has this format, where `T` is the time of the attempt in Unix seconds and `HEX` is the lowercase hexadecimal HMAC-SHA256 of `T`, a dot and the exact request body, keyed with the whole signing secret (including `whsec_`):

```
X-Xtrakt-Signature: t=T,v1=HEX
signed string:      T.BODY
```

To verify a request, recompute the HMAC over the timestamp, a dot and the raw body as received, and compare it with `v1` in constant time. The signature is rebuilt at each attempt, with a new timestamp.

The secret's description in the dialog names the same header: *Use this secret to verify the X-Xtrakt-Signature header on incoming webhook requests (HMAC-SHA256 over `<timestamp>.<body>`).*

Delivery and retries

- Only **Active** endpoints subscribed to the event, in the event's environment, receive it.
- Your server must answer within 10 seconds. Redirects are not followed. Any `2xx` answer counts as delivered.
- A delivery that fails with a `5xx`, `408` or `429` answer, a timeout or a connection error is tried again, with a growing delay between 10 seconds and 10 minutes, up to **5 attempts** in all. Any other `4xx` answer is final and not retried.
- Every attempt is listed in Webhook Request Logs.

Buttons and actions

Button	What happens	Who	Reversible?
+ New endpoint	Opens **Create webhook endpoint**.	Tenant administrators	—

Button	What happens	Who	Reversible?
+ Add header	Adds a header row. **×** removes one.	—	—
Save endpoint (**Saving…** while it runs)	Checks the fields, then creates or updates the endpoint. On create, and on an update that changed the URL, shows the signing secret.	Tenant administrators	Yes — edit again.
Copy / **Copied!**	Copies the signing secret.	—	—
Done	Closes the secret panel. The secret cannot be shown again.	—	—
Cancel	Closes the dialog without saving.	—	—
Edit	Opens **Edit webhook endpoint** with the endpoint's values.	Tenant administrators	—
Delete	Asks **Delete endpoint** — *Endpoint … will stop receiving events. This operation is irreversible.* — then **Yes, delete** removes it.	Tenant administrators	No

This screen has no test button. A test event can be sent through the API (see the developer portal) or from an editor connected to the MCP server.

Common problems

Message	Cause	What to do
Something went wrong. Please try again.	The service refused the endpoint — for example the URL points to a private or local address — or the request failed.	Use a public HTTPS URL and save again.

Message	Cause	What to do
No endpoints yet. Create your first to start receiving events.	No endpoint exists — or the list could not be loaded, which the screen does not distinguish.	Check that you are a tenant administrator.
Your server rejects every request as unsigned	It looks for a header other than `X-Xtrakt-Signature` — for example `X-Cogneris-Signature`, which earlier versions of the dialog named.	Read `X-Xtrakt-Signature` (see How each request is signed).
Signatures do not match	The body was re-serialised before hashing, or the URL was changed and the old secret is still in use.	Hash the raw body; use the secret shown after the last URL change.
Loading endpoints…	The list is loading.	—

API Request Logs

API Request Logs — *Every API call billed to this tenant.* A read-only list of the calls your company made to the Cogneris API, newest first.

Fields

Field	What it does	Format / values
Environment (filter)	Shows only the calls of one environment. Applied by the service.	**All environments**, **Sandbox**, **Production**, **Unknown origin** (calls with no recorded environment)
Created at	When the call was made.	Date and time
IP	The address the call came from.	Text, or `-`
Duration	How long the call took.	Milliseconds

Field	What it does	Format / values
Environment	The environment of the key used.	**Sandbox**, **Production** or **Unknown origin** badge

View details opens **Call details**: **Created at**, **Duration**, **IP**, **Environment**, and **Response** — the details the service recorded for the call, as formatted JSON.

Buttons and actions

Button	What happens	Who	Reversible?
Refresh (**Loading…** while it runs)	Loads the list again.	Tenant administrators	—
View details	Opens **Call details**. Close it with ✕ or by clicking outside.	Tenant administrators	—

Common problems

Message	Cause	What to do
Could not load API logs. Please try again. (with the HTTP status when there is one)	The list could not be read. `403` means your profile is not a tenant administrator.	Use **Refresh**, or ask an administrator.
No API calls yet.	No call matches the filter.	Choose **All environments**.

Webhook Request Logs

Webhook Request Logs — *Every webhook delivery billed to this tenant.* A read-only list of the webhook deliveries Cogneris made to your endpoints, one row per attempt.

Fields

Field	What it does	Format
Created at	When the delivery was made.	Date and time
Webhook	The webhook the delivery belongs to, as the service recorded it.	Text, or `-`
Event	The event delivered.	Event name

View details — shown on rows that have a delivery record — opens **Delivery details**:

Field	What it shows
Success / Failure	Whether the endpoint answered with a `2xx`.
HTTP status	Your server's status code, or `-` when it did not answer.
Duration	Time to the answer, in milliseconds.
Attempt	The attempt number (see Delivery and retries).
URL	The URL called.
Event	The event name.
Error message	Shown when the attempt failed without an answer or reached the attempt limit, for example `webhook_timeout`, `webhook_dispatch_failed` or `webhook_retries_exhausted`.
Request body	The JSON sent.
Response body	What your server answered, or `-`.

Buttons and actions

Button	What happens	Who	Reversible?
Refresh (**Loading…** while it runs)	Loads the list again.	Tenant administrators	—
View details	Opens **Delivery details**. Close it with ✕ or by clicking outside.	Tenant administrators	—

Common problems

Message	Cause	What to do
Unable to load delivery details.	The delivery record could not be read.	Close the dialog and try again.
No webhook deliveries yet.	No delivery was made — or the list could not be loaded, which the screen does not distinguish.	Use **Refresh**; check that you are a tenant administrator.

Related

- MCP — developer tokens and connecting an editor.
- Vision administration overview

Last reviewed 2026-10-23 · Version 1.0.0 · Owner: product-documentation

MCP

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

The Cogneris MCP server lets an AI assistant in a developer's editor — VS Code, Cursor, Claude Code or any other client of the Model Context Protocol (MCP) — work with your company's Cogneris account: read the API documentation, and create API keys, webhook endpoints and classifiers in plain language. This page gives the server address, ready-made editor configuration, and the developer tokens that keep an editor connected.

How to get here	Vision menu › **MCP**. The old Playground chat address now opens this page.
Who can use it	Every signed-in member can open the page. Creating, listing and revoking developer tokens is limited to tenant administrators (`customer-admin` , `customer-tech` , and Cogneris operators inside their own company), like API keys. What a connected editor may do is set by the token's permissions or, without a token, by your own account.
Content status	From source; not observed (`vision-fe` `0b724a0` , `mcp-be` `863cdb5` , `billing-be` `12ce77c` , 2026-10-22)

The page title is **MCP**, with the line *Connect the Cogneris MCP server to VS Code, Cursor or Claude Code and build with the Cogneris API from your editor: create API keys, webhooks, classifiers, workflows and PDF fill templates.*

Server URL

Server URL shows the address editors connect to, with **Copy** / **Copied**. The hint says *This server is in your account's region. Use the same URL in every editor.*

There is one server per data region, for example `https://mcp-eu.cogneris.ai/` and `https://mcp-us.cogneris.ai/`. The page picks the one for your company's region.

Connect your editor

Connect your editor has one tab per editor: **VS Code**, **Cursor**, **Claude Code**. Each tab shows a snippet with the server URL already filled in, and **Copy** / **Copied**. The snippets below use the European address as an example.

VS Code

Save this as .vscode/mcp.json in your project. VS Code asks for your developer token the first time and keeps it in its secret storage.

```
{
  "inputs": [
    {
      "id": "cogneris-token",
      "type": "promptString",
      "password": true,
      "description": "Cogneris developer token"
    }
  ],
  "servers": {
    "cogneris": {
      "type": "http",
      "url": "https://mcp-eu.cogneris.ai/",
      "headers": { "Authorization": "Bearer ${input:cogneris-token}" }
    }
  }
}
```

The tab also has the link **Install in VS Code**: *The one-click install signs you in through the browser instead of asking for a token.* (see Signing in without a token).

Cursor

Add this to ~/.cursor/mcp.json and replace the placeholder with your developer token.

```
{
  "mcpServers": {
    "cogneris": {
      "url": "https://mcp-eu.cogneris.ai/",
```

```
      "headers": { "Authorization": "Bearer <YOUR_COGNERIS_TOKEN>" }
    }
  }
}
```

Claude Code

Run this in your terminal and replace the placeholder with your developer token.

```
claude mcp add --transport http cogneris\
  https://mcp-eu.cogneris.ai/ \
  --header "Authorization: Bearer <YOUR_COGNERIS_TOKEN>"
```

The page shows the same command on one line.

Any other MCP client connects the same way: an HTTP MCP server at the **Server URL**, with the header `Authorization: Bearer` followed by the developer token.

Signing in without a token

Prefer to sign in through the browser? — *Editors that support OAuth can connect without a token: they open the Cogneris sign-in page and use your account. That session lasts 1 hour, while a developer token keeps you connected for up to 90 days.*

When an editor connects without a token, the server answers that sign-in is required and tells the editor where to sign in; an editor that supports OAuth then opens the Cogneris sign-in page. The editor acts with your own account until the session ends, after one hour.

Developer tokens

Developer tokens — *A developer token connects your editor to Cogneris on your behalf. It expires on its own, and you can revoke it at any time.*

A developer token is a special API key:

- It is tied to the user who creates it; the server records every action under that user.
- It always expires — after 30 or 90 days. The service refuses a developer token without an expiry or with one longer than 90 days.
- It carries the permissions chosen at creation.
- It is created in your company's default production environment, so it starts with `xtkt_live_`.

- It works only with the MCP server. The MCP server refuses every other API key, and developer tokens do not appear on the **API Keys** page (see Developers).

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Token name	A label, for example the computer it is used on.	Yes	Up to 120 characters. Placeholder *Work laptop*.	Empty	**Generate token** stays disabled while empty.
Expires in	How long the token works.	Yes	**30 days**, **90 days**	**90 days**	—
Permissions	What a connected editor may do. Each option shows its label and its scope.	At least one	See Permissions	**Read your resources** and **Create and change resources**	*Choose at least one permission.*

Permissions

Option	Scope	What it allows
Read your resources	`mcp.read`	Read tools: who you are, the documentation and API reference, classifiers and their training status, webhook event types, PDF fill templates and fill jobs, downloads of filled PDFs, and your usage summary.

Option	Scope	What it allows
Create and change resources	`mcp.write`	Create and train classifiers and add samples to them; register PDF fill templates, upload source PDFs and start fill jobs.
Manage API keys	`apikeys.manage`	List, create and revoke your company's API keys. Never implied by **Create and change resources**.
Manage webhooks	`webhooks.manage`	List, create, update, disable, delete and test webhook endpoints, replace their signing secret, and list and replay deliveries. Never implied by **Create and change resources**.

A tool whose permission the token lacks is not offered to the editor and refuses to run.

The token list, **Active tokens**:

Field	What it shows	Format
Name	The token name.	Text
Token	The last four characters of the token.	`••••` and four characters
Expires	The expiry date.	Date
Last used	Date of the last use, or **Never**.	Date
Actions	**Revoke**.	—

Only tokens that are neither revoked nor expired are listed.

Buttons and actions

Button	What happens	Who	Reversible?
Generate token (**Generating…** while it runs)	Creates the token and shows it once, with *Copy this token now. For your security, it will not be shown again.*	Tenant administrators	Revoke the token to undo.
Copy / **Copied**	Copies the new token.	—	—
Done	Hides the token. It cannot be shown again.	—	—
Revoke	Asks **Revoke developer token?** — *Any editor using "…" loses access within a minute. This cannot be undone.* — then **Yes, revoke** revokes the token; **Cancel** keeps it.	Tenant administrators	No

The MCP server reuses a successful token check for up to 60 seconds, so an editor that is already connected can keep working for up to a minute after **Revoke**.

Common problems

Message	Cause	What to do
The token could not be created. Please try again.	The service refused the request — for example your profile is not a tenant administrator.	Ask a tenant administrator to create the token.
The token could not be revoked. Please try again.	The revocation failed.	Try again.
You have no active developer tokens yet.	No active token — or the list could not be loaded, which the page does not distinguish.	Check your profile with an administrator.

Message	Cause	What to do
The editor keeps asking to sign in	The token expired or was revoked, or the editor signed in through the browser and the one-hour session ended.	Create a new token and update the editor's configuration.
Loading tokens…	The list is loading.	—

What you can ask

What you can ask — *Once connected, ask your editor's assistant in plain language. For example:*

Area	Example on the page
API keys	*Create a sandbox API key for my checkout service.*
Webhooks	*Send every document event to a webhook at https://example.com/hooks.*
Classifiers	*Create a classifier for invoices and receipts, then test it in the Playground with this PDF.*
Workflows	*Build a workflow that extracts the ID card, checks the face match and sends doubtful cases to manual review.*
Extraction	*Extract this invoice with my Invoice classifier and show me the fields as JSON.*
PDF fill templates	*Register a fill template for this PDF form and fill it with the data from my last extraction.*

Limits

Limit	Value
Developer token lifetime	30 or 90 days, chosen at creation; never more than 90.
Browser sign-in session	1 hour.
Revocation delay	Up to 60 seconds for an editor already connected.
API keys created from an editor	Sandbox by default. A production key is created only when the request names the production environment and confirms it explicitly.
Source PDF upload	10 MB.
Webhook deliveries per page	Up to 25.
Documentation read in one call	Up to 6,000 characters; longer pages are read in parts.
Request rate	Decided by the service; above it the server answers that the request was rate-limited.

At this version the server's tools cover API keys, webhooks, classifiers, workflows and their runs, Build AI, PDF fill templates and jobs, usage and documentation. It has no tools for portal forms or Word templates, and the page's examples no longer mention them.

Related

- Developers — API keys and webhook endpoints from the Vision screens.
- Vision administration overview

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Template administration overview

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Template is the Cogneris document-generation product. You upload a Word document (.docx) that contains placeholders, say where the value of each placeholder comes from, publish a version, and Cogneris renders it to PDF — for a Workflow process or for your own application through the API.

How to get here	The Template product in the platform menu, or the Template application on its own (its window title is **Cogneris Templates**). It opens on **Dashboard**.
Who can use it	**Templates** and **Playground**, and every action in them: the tenant-management profiles — customer administrator (`customer-admin`), customer technical administrator (`customer-tech`), and the Cogneris operators (`cogneris-admin` , `cogneris-tech`). Deleting a template additionally needs the tenant administrator role (`tenant.admin`). **Dashboard**: any signed-in member of the tenant. Checked by the service; hiding a menu entry is not the control.
Content status	From source; not observed (`template-fe` `ed4ab73` , `template-be` `cc0f2b0` , 2026-10-22)

Menu and routes

The menu has three entries. The breadcrumb starts at **Template** (which links to **Dashboard**); a template page reads **Template** › **Templates** › the template's name.

Menu entry	Address	Screen	Chapter
Dashboard	`/dashboard` (also `/`)	Counters and quick actions	Dashboard below
Templates	`/templates`	The list of templates; create and delete	Templates
— (open a template card)	`/templates/` + template id	Versions, upload, publish, detected keys, Redaction policy	Templates
— (no menu entry or link)	`/templates/` + template id + `/bindings`	**Field sources**: where each placeholder's value comes from	Field sources
Playground	`/playground`	Render a published template to PDF in the browser	Render playground

An address that matches no screen shows *Page not found* with a **Go home** button.

When Template is shown inside the platform, it hides its own side menu, top bar and breadcrumb, and the platform shows the same entries and trail instead.

From a .docx to a rendered document

1. **Prepare the .docx.** Write the document in Word and put a placeholder wherever a value goes — two opening braces, a key, two closing braces (see Placeholder syntax). The Word add-in inserts placeholders from the list of fields your processes can supply.
2. **Create the template and upload the .docx** — Templates. Each upload is a new version; Cogneris lists the keys it found.
3. **Say where each value comes from** — Field sources. Each placeholder is bound to a document field, a form answer, a process value or the cross-check result, with a rule for when the value is missing.
4. **Publish a version** — Templates. Only the published version is rendered; drafts are never served.

5. **Render** — try it in the Render playground, then render from a Workflow phase or from the API (below).

Dashboard

The landing screen. It loads once when you open it.

Counters

Counter	What it counts
Active templates	Templates of your tenant that are not deleted, published or not.
Published	Templates that have a published version.
Archived	Templates that were deleted — the service keeps them archived (see Delete templates).
Renders (7d)	PDFs rendered for your tenant in the last 7 days, from every caller (Playground, Workflow, API). Shown as `+N`, or `0`.

A counter shows `–` while it has no value.

Quick actions

Tile	Where it goes
New template — *Upload a .docx with … placeholders. We detect the keys and let you render PDFs.*	**Templates**. Use **+ New template** there.
All templates — *Browse, search, archive, or open any template to manage versions and keys.*	**Templates**.
Render playground — *Try a render in the browser with an API key: fill the keys and generate a PDF.*	**Playground** — see Render playground.

Tile	Where it goes
Publish a version — *Only published versions are reachable from the M2M render endpoint. Promote a draft from any template.*	**Templates**, showing only the templates with no published version (**Show all** returns to the full list).

Common problems

Message	Cause	What to do
We couldn't load your dashboard stats. with **Try again**	The service did not answer the counters request.	Select **Try again**.

How a published template is used

Only the **published** version of a template is ever rendered. A template is addressed by its **code** — the slug Cogneris made from the name when the template was created (shown under the name on every card), which never changes. Each render counts toward the tenant's monthly render quota (5,000 renders a month unless your contract sets another value); a render served from the cache does not.

Workflow — the Generate document phase

The Generate document phase names the template by its **Template code**. At run time it sends the whole case — documents and their extracted fields, form answers, process data, the cross-check result — and the template's own field sources decide which value each placeholder takes:

- When every placeholder that matters can be filled, the service renders the PDF and returns it to the run.
- When a placeholder whose rule is **Send for review** cannot be filled, the service renders nothing and answers *needs review*, with the list of missing keys and why each is missing. The phase's **Print even with gaps** option makes the service render anyway, printing the fallbacks.

The details of what makes a value missing are in Field sources — at render time.

Your application — the render API

`POST /templates/` + code + `/render` returns the PDF of the published version, filled with the values you send. You send the values yourself; the template's field sources are not used on this route.

Part	What it is
`X-Xtrakt-Api-Key` header	A tenant API key for the **production** environment. Sandbox keys are refused.
Proof of an approved caller	The request must also carry a body signature (`X-Xtrakt-Timestamp`, Unix seconds within 5 minutes of the server clock, and `X-Xtrakt-Signature`, an HMAC-SHA256 described below), or come from a Cogneris service. Browser calls are refused unless the environment allow-lists that browser origin.
Body	A JSON object with `values` (a list of key/value pairs) and the optional `cache`, `strict` and `missingPlaceholder` (example below).
Answer	`application/pdf`, named code + `-v` + version number + timestamp + `.pdf`, with the headers `X-Xtrakt-Template-Version`, `X-Xtrakt-Render-Elapsed-Ms`, `X-Xtrakt-Render-Id` and `X-Xtrakt-Cache-Hit` (`1` or `0`).
Rate limit	10 requests a minute per API key.

```
{
  "values": [
    { "key": "nome", "value": "Maria Silva" },
    { "key": "valor", "value": "1.250,00" }
  ],
  "cache": false,
  "strict": true,
  "missingPlaceholder": ""
}
```

- A placeholder you do not send is printed as `missingPlaceholder` (empty by default), so no braces are left in the PDF.
- `strict` (default on) refuses a render that leaves out a required key. Uploads create every key as not required, so this check does not refuse anything today.

- `cache` (default off) reuses the PDF rendered earlier for the same version and exactly the same values, for 30 days.

The signature is the lowercase hexadecimal HMAC-SHA256 of the text below, with a signing secret configured on the service. How that secret is issued is not shown in the source; ask Cogneris.

```
timestamp.METHOD.path.code.apiKey.bodySha256
```

`METHOD` is in capitals, `path` is the request path, and `bodySha256` is the lowercase hexadecimal SHA-256 of the raw request body.

Answer	Cause
401 *render endpoint requires an approved origin or internal service token*	No valid signature, and not an allow-listed origin.
401 *X-Xtrakt-Api-Key header is required / invalid tenant api key*	No key, or a key that is revoked, unknown or not for production.
400 *template '…' not found*	Unknown code, or a deleted template.
400 *template has no published version*	Publish a version first.
400 *monthly quota exceeded (…/…)*	The month's render quota is used up.
400 *template source (…) exceeds the synchronous render limit (…)…*	The published .docx is larger than 25 MB (the default limit).
429	More than 10 requests in a minute.

Related

- Templates — create, upload, versions, publish, delete.
- Field sources — where each placeholder's value comes from.
- Render playground — try a render.
- Word add-in — prepare the .docx in Word.
- Generate document — the Workflow phase that renders a template.

- Vision administration overview — the document templates (classifiers) whose extracted fields feed document sources.

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Templates

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Create document templates, upload their .docx versions, see the placeholders Cogneris found, publish the version that is rendered, and delete templates you no longer use.

How to get here	**Templates** in the Template menu. A template page: select a card in the list.
Who can use it	Tenant-management profiles (customer administrator, customer technical administrator, Cogneris operators). Deleting also needs the tenant administrator role (`tenant.admin`). Checked by the service.
Content status	From source; not observed (`template-fe` `ed4ab73` , `template-be` `cc0f2b0` , 2026-10-22)

Template list

The header reads **Templates** with *… total* under it — the number of templates that match the search, archived (deleted) ones excluded.

Each template is a card with its name, a status chip, its **code**, its description (when it has one) and *N* **Versions**. Cards are ordered by last change, newest first.

- **The list shows at most 20 templates** — the 20 most recently changed that match the search — and has no next page. The total counts all of them. Use the search to reach the others.
- **The status chip** reads **Published** when the template has a published version, **Draft** when it has none, and **Archived** for an archived template. (The service's own template

status is only *Active* or *Archived*; whether a version is published is recorded on the version.)

While the list loads, six grey placeholder cards are shown. With no templates (or no match), the list shows *No data*.

Only templates without a published version. Arriving from the Dashboard's **Publish a version** tile, the list shows only the templates that have no published version, under the note *Showing templates without a published version.* **Show all** returns to the full list. This view looks through the 100 most recently changed templates.

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Search by name or code	Filters the list as you type (after a short pause). Matches any part of the name or the code, ignoring case.	No	Text	Empty, or the value of `?search=` in the address	—
Card check box (*Select template …*)	Selects the card for deleting. Selecting it does not open the template.	No	On / off	Off	—

Buttons and actions

Button	What happens	Who	Reversible?
Copy link	Copies the address of this page, search included, to the clipboard. The label changes to **Copied!** for about two seconds.	Anyone who can open the page	—

Button	What happens	Who	Reversible?
+ New template	Opens the New template dialog.	Tenant-management profiles	—
A card	Opens the template page.	Tenant-management profiles	—

Create a template

+ New template opens a dialog titled **New template**.

Field	What it does	Required	Format / values	Default	Validation messages
Name	The template's display name. The service also makes the template's **code** from it.	Yes	Text; spaces at both ends are removed	Empty	**Save** stays disabled while it is empty.
Description	Free text shown on the card.	No	Plain text. HTML markup is refused — a less-than sign directly followed by a letter, `/`, `!` or `?`, also when the sign is written as an HTML entity. A	Empty	*description must not contain HTML markup*

Field	What it does	Required	Format / values	Default	Validation messages
			less-than sign followed by a space or a digit, and an e-mail address in angle brackets, are accepted.		

Button	What happens	Who	Reversible?
Save	Creates the template, shows *Template created* and opens its page. The template has no versions yet.	Tenant-management profiles	Delete it (archives it).
Cancel, or a click outside the dialog	Closes the dialog. What you typed is kept until you leave the page.	—	—

How the code is made. Accents are removed, the name is put in lowercase, and every run of characters other than `a` – `z` and `0` – `9` becomes one hyphen (*Fatura Mensal V1* gives `fatura-mensal-v1`). If the code is taken — also by a deleted template — `-2`, `-3` … is added. An empty result becomes `template`. The code cannot be changed afterwards; the name and description can, with **Edit details** on the template page.

Template page

The header shows the template's name, and under it the code and *N* **Versions**. While the template loads, the page shows *Loading…*.

Button	What happens	Who	Reversible?
← Back	Returns to the list.	—	—

Button	What happens	Who	Reversible?
Edit details	Opens a dialog with **Name** and **Description** (same rules as in Create a template) and the note *The code … never changes, so callers keep working.* **Save** stores them and shows *Template updated*; a blank description removes it.	Tenant-management profiles	Yes: edit again.
Field sources	Opens the template's field sources.	Tenant-management profiles	—

The left column lists the **Versions**, with the upload area under them. The main area shows the version you select; when the page opens it shows the published version, or the newest one when none is published. With no versions, it shows *Pick a version*.

Upload a version

The upload area reads **Drop a .docx file or click to choose**. Drop a file on it, or click it to choose one; the upload starts at once. Only the first file is used when you drop several.

Above it, **Notes for the next version (optional)** — up to 1,000 characters stored with the version you upload next and shown on its row in the timeline. The box is cleared after a successful upload.

Rule	Limit	Message
A file must be sent	—	*file is required*
Not empty	—	*file is empty*
Size	50 MB	*file exceeds the maximum allowed size of 50 MB*
Extension	`.docx` only — the file chooser offers only .docx, and the service checks the name	*only .docx files are allowed*

Rule	Limit	Message
Real .docx content	Must be a ZIP (Word) archive	*file is not a valid .docx (zip) archive.*
Number of parts inside the archive	512	*document archive has too many entries (limit 512).*
Uncompressed size	200 MB	*document archive uncompressed size exceeds the limit (200 MB).*
Compression ratio	100 to 1	*document archive compression ratio exceeds the limit (100:1).*
Text parts read for placeholders	64 parts, 2 MB each, 8 MB in all	The upload fails (the service's message).

On success:

- **New content** — *Version … uploaded*. The new version gets the next number and is selected; it is a draft (not rendered) until you publish it.
- **Same content as an earlier version** (byte-for-byte identical) — *Content identical to an existing version — new version tracked as duplicate*. A version is still added with the next number, the stored file is reused, and its keys are copied from the earlier version. In the timeline it carries the chip **Duplicate of v…** with the number of the version it repeats.

A file larger than 25 MB can be uploaded but cannot be rendered with the default settings (see overview — render API).

Placeholder syntax

The service reads the main text, the headers, the footers, the footnotes and the endnotes of the .docx, and recognises a placeholder written as two opening braces, a key and two closing braces:

```
{{nome}}
{{ titular.cpf }}
{{documentos.documentName}}
```

- The key may use letters (A–Z, a–z), digits, `_`, `-` and `.`. Spaces inside the braces are ignored. Anything else — accented letters, spaces inside the key — is not a placeholder and is left in the text as typed.
- Keys are case-sensitive when they are detected: `Nome` and `nome` are two keys.
- Word often splits text into pieces when you edit it; the service joins them before looking, so a placeholder typed in two goes is still found.
- A dot names a field inside a group. A key such as `documentos.documentName` is how a repeating block is written.

Versions and publishing

Each version row shows **v** + number, the file name, its size, and when it was uploaded (in the language of the interface), and its note when it was uploaded with one. The published version carries a **Published** chip. Select a row to show that version's keys.

Button	What happens	Who	Reversible?
Publish this version (on every version except the published one)	Makes this version the one rendered for the template's code, at once and without a confirmation. Shows *Version published*. Every render after this — Playground, Workflow, API — uses it. The version published until then becomes **Archived**: a template has one published version at a time. Keys of the new version that have no **Required** / **Default value** of their own take them from the version it replaces.	Tenant-management profiles	Yes: publish another version. Any older version, archived ones included, can be published again.

The template's field sources belong to the template, not to a version: a new version keeps every source whose placeholder it still contains.

Keys detected

The selected version's heading shows **v** + number, its file name, and a **Published** chip when it is the published one. Under it, **Keys detected** (*count*) lists the placeholders the service found, in alphabetical order:

Column	What it shows
Key	The placeholder's key.
Occurrences	How many times it appears in the document.
Type	Always **string**: every key is stored as text.

With no placeholders, the table reads *No keys found*.

Redaction policy

A card under the keys, **Redaction policy** — *Publish the tenant's default Redaction policy for this template.* The policy itself is kept and enforced by the Vision service; this card only shows its state and publishes the tenant default.

State chip	Meaning
Not published	The template has no Redaction policy.
Draft	A policy version was started but not published.
Published · Version …	Published, with the number of the newest published policy version. Under it: *… categories · …-hour retention*.

Button	What happens	Who	Reversible?
Publish policy (when nothing is published)	Asks *Publish Redaction policy? — This template will use every category allowed by the tenant policy.* Confirming with **Publish policy** creates a draft from the tenant's live policy, validates it and publishes it; *Redaction policy published* confirms. **Cancel**, Esc or a click outside closes the question.	Decided by the Vision service	Decided by the Vision service.

Button	What happens	Who	Reversible?
Resume publishing (when a draft exists)	Same question; validates and publishes the existing draft.	Decided by the Vision service	Decided by the Vision service.

If the state cannot be read, the card shows *The Redaction policy could not be loaded.* with **Retry**, and no publish button.

Delete templates

Select one or more cards. A bar appears with *… template selected* (or *… templates selected*) and three buttons:

Button	What happens	Who	Reversible?
Select all visible	Selects every card on the screen.	—	—
Archive	Asks *Archive templates? — This archives … templates: they leave this list and can no longer be rendered. Their versions are kept.* Confirm with **Archive**; **Cancel**, Esc or a click outside closes the question. While it runs the button reads **Archiving…**.	Tenant administrator (`tenant.admin`)	Not from the screen (see below).
Clear selection	Unselects every card.	—	—

What happens in the service: the template is **archived**, not erased. It leaves the list and the Playground, every render of its code answers *template '…' not found*, and it is counted under **Archived** on the Dashboard. Its versions and render history are kept, and its code is never given to another template. The screen offers no way to restore it. Deleting a template that is already archived succeeds and changes nothing.

Each template is deleted separately, so one result can mix outcomes:

Message	Meaning
Archived … templates	That many were archived.
Templates not archived: …. Only a tenant admin can archive templates.	You lack the tenant administrator role. These are not offered for retry.
Failed to archive some templates — click to retry	Others failed for another reason. They stay selected; click the message to try them again. If that also fails: *Failed to archive some templates (retry failed).*

Common problems

Message	Cause	What to do
The list is empty after the Dashboard's **Publish a version** tile	Every template already has a published version.	Select **Show all**.
A template you know exists is not in the list	The list shows only the 20 most recently changed matches.	Search by its name or code.
An error message with *— click to retry* when the list loads	The list request failed; the message is the service's.	Click the message to load the list again with the same search.
description must not contain HTML markup	The description contains markup.	Remove the tag-like text.
template not found	The template was deleted, or belongs to another tenant.	Return to the list.

Message	Cause	What to do
version not found for this template	The version is not part of this template.	Reload the page.
An upload message from the table in Upload a version	The file broke a limit.	Save the document again as a Word .docx and check its size.
A placeholder is missing from **Keys detected**	Its key uses a character outside letters, digits, `_`, `-`, `.`, or it is in a part the service does not read (for example a text box outside the main text, headers, footers, footnotes and endnotes).	Rename the key and upload a new version.
Duplicate of v… on a version	You uploaded a file identical to that earlier version.	Nothing to do; publish either.

Related

- Template administration overview
- Field sources — bind each detected key.
- Render playground — check the published version.
- Word add-in — insert placeholders in Word.
- Generate document

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Field sources

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Field sources says, for every placeholder of a template, where its value comes from when a Workflow process renders the document, and what to do when that value is missing. These settings are called *bindings*.

How to get here	**Templates** → open a template → **Field sources** at the top of the template page. The breadcrumb reads **Dashboard › Templates › *template name* › Field sources**; the template's name takes you back to its versions.
Who can use it	Tenant-management profiles (customer administrator, customer technical administrator, Cogneris operators). Checked by the service.
Content status	From source; not observed (`template-fe` `ed4ab73` , `template-be` `cc0f2b0` , 2026-10-22)

Bindings are used only by the case-driven render — the Generate document phase of Workflow. A render through the public API or the Render playground receives its values directly and ignores them.

What the screen shows

The heading **Field sources** and: *Where the value of each placeholder comes from in the published version. A key without a source is left blank in the generated document — or sends the process for review, depending on the rule.*

While loading: *Loading…*. Then:

- **Only the published version is compared.** The rows are the keys of the template's published version plus every key that already has a binding. With no published version (or one with no placeholders) the screen shows *This template does not have a published version with placeholders yet.* and no table.
- *… key(s) without a source: …* — shown above the table when any key has no binding. Those rows are highlighted.
- *Bindings for keys that no longer exist in the document (…) — saving removes them.* — bindings whose key is not in the published version. They are not rows of the table (there is nothing left to bind), and the next **Save** removes them.
- Under a repeating block, its own sub-keys, read-only: *covered by* and the block's key.
- Under the table, one line per source group that has nothing to offer yet, with the service's explanation of how to configure it (see Source groups).

Bindings belong to the **template**, not to a version. When you publish a new version, every binding whose key the new version still contains keeps working; bindings for keys you removed become the "no longer exist" kind above.

Fields

One row per key, sorted alphabetically.

Field	What it does	Required	Format / values	Default	Validation messages
Placeholder	The key, as detected in the .docx.	—	Read-only	—	—
Source	Where the value comes from. Grouped list of every field your processes can supply; entries that may be empty in a	No	One field from the source groups	**— without a source —** for a new key	See Save

Field	What it does	Required	Format / values	Default	Validation messages
	case end with *(not always filled in)*. — **without a source —** removes the binding.				
If missing	What the render does when the source has no value in the case. Shown only once a source is chosen.	Yes, when bound	**Send for review**, **Leave blank**, **Use default value** — see What happens at render time	**Send for review**	—
Repeats	Whether the key prints one value or repeats a table row. Shown only once a source is chosen.	Yes, when bound	**Once**, **Repeatable block** — see Repeating blocks	**Once**	*binding '…': a repeating block needs source 'document' (got '…')*
Required	Marks the key as one the document cannot go without —	No	Checkbox	Off	—

Field	What it does	Required	Format / values	Default	Validation messages
	see What happens at render time.				
Default value	What prints when the key has no value: for a key with no binding, and for **Use default value**. Hint: *Printed when there is no value*. Leave empty for none.	No	Free text, up to 500 characters	Empty	*….: the default value is longer than 500 characters.*

Required and **Default value** are stored on the **published version**'s keys, so they appear only once a version is published (*Publish a version to mark keys as required or give them a default value.* otherwise). A new version you upload, or publish, keeps them for every key it still has. **Save** sends them after the sources; *The sources were saved, but the required / default settings could not be.* when only that part fails. With **Use default value** chosen and no default set, the screen warns *“Use default value” prints nothing until a default is set: ….*

A binding made outside this screen — through the API or the Word add-in — that does not match an entry of the list shows **— without a source —** in **Source**, but it is kept and saved unchanged unless you pick another source. The same goes for a binding's **transform** (a presentation format such as upper case, or a date or number format), which this screen does not show or edit: it is kept as it was.

Source groups

The **Source** list comes from the service, which keeps it for 5 minutes per tenant — a field added in Vision or Portal can take that long to appear. The fixed groups (**Process**, **Cross-**

check), their fields and the explanations of empty groups are shown in your language; classifier and form groups carry the names your tenant gave them.

Group	Fields offered	Value at render time
One group per Vision document template (classifier) of your tenant, under the classifier's name	Every field defined in the classifier's schema. All are *(not always filled in)*: a document is only in a case if it was uploaded, and a field only has a value if extraction read it.	The field's extracted value, from the case's documents that were classified with that classifier.
One group per Portal form, under the form's name	Every answer field of the form (fixed texts are left out), labelled with the field's label. A field marked required in the form is not flagged *(not always filled in)*.	The answer of that name in the case's form data.
Process	**Process identifier** (`identifier`), **Requester name** (`requester.name`), **Today's date** (`today` , always present).	The process data the run sends.
Cross-check	**Cross-check decision** (`decision`), **Number of divergences** (`summary.mismatch`), **Number of checks** (`summary.total`).	The cross-check result the run sends.

A binding saved earlier as `crosscheck.findingsCount` — a field the list used to offer but that no run ever sent, so it always printed empty — is read as **Number of divergences** (`summary.mismatch`) and shows as that entry in **Source**.

When a group cannot be filled, it still appears, as a line under the table: for example, no classifiers yet (*This tenant has no classifiers yet…*), a classifier with no fields defined, no forms, a form with only fixed text, or the Vision or Portal service not answering.

The list offers each key namespaced by its group (for example a classifier *Documento de Identidade — Beneficiário* gives keys starting `documentoIdentidadeBeneficiario.`), so two classifiers with a field of the same name can be told apart. Picking a document field also records which classifier the value must come from.

What happens at render time

When the Generate document phase renders a template, the service takes the **published** version's keys and the template's bindings, and works out each key's value from the case:

1. **Key with no binding** — prints its **Default value** when it has one. Otherwise it prints empty and is listed as missing (`no_binding`); that stops the render only when the key is **Required**.

2. **Key with a binding** — the value is looked up in the binding's source:

   - **form**, **process**, **cross-check**: the named field.
   - **document**: the documents of the case that fill the binding's role. The role is matched most-trustworthy first — the checklist slot the document was uploaded into, then the classifier (Vision template) that processed it, then its classified document type. The first matching document that has a non-empty value wins.

3. **The value is found** — it is printed (with its transform, if the binding has one).

4. **The value is not found** — the **If missing** rule decides:

Rule	Prints	Stops the render?
Send for review (default)	Nothing	**Yes.** No PDF is produced; the service answers *needs review* with the missing keys, unless the phase has **Print even with gaps** on.
Leave blank	An empty value	No.
Use default value	The key's **Default value**	No — except when no default is set and the key is **Required**: then it stops the render like **Send for review**.

   Each missing key is reported with a reason: `role_not_present` (no document of the case fills the role), `field_not_present` (the document is there but the field was not read), `document_failed` (every matching document failed in an earlier phase), or `unknown_source` .

The missing list, with each key's reason and whether it blocked, is returned to the Workflow run — see Generate document.

Repeating blocks

Repeatable block turns a binding into a list: the service sends one row per document of the case that fills the binding's role, and the table row of the .docx that holds the block's sub-keys is printed once per document. Write the row once in Word, with sub-keys named after the block's key:

```
{{documentos.documentName}}   {{documentos.documentType}}
{{documentos.cpf}}
```

- The block is bound on the prefix (`documentos` above). It covers that key and every key that starts with it followed by a dot, which appear under it as *covered by* `documentos`.
- In each row, the sub-key is a field extracted from that document; `documentName` and `documentType` are the document's name and document type.
- **Only document sources can repeat.** Choosing **Repeatable block** on a form, process or cross-check field is refused on save.
- **A block that matches no document disappears.** This is how a conditional is written: "this clause only when there is a second holder" is a repeating block on that role. An empty block is recorded as missing (`role_not_present`) but never stops the render.
- A sub-key cannot also have a binding of its own: *binding '…': already covered by the repeating block '…'*.

The screen lists the keys found in the document, which for a block are the sub-keys (`documentos.cpf`), not the prefix. The prefix row appears only once a binding exists for it — for example one created with the Word add-in or the API.

Save

Save (**Saving…** while it runs) sends the **complete** set of bindings shown: anything not in it is removed. On success:

- *… binding(s) saved. All keys are covered.*
- *… saved — … key(s) still without a source.* — the save worked; the count is the published version's keys that still have no binding (sub-keys of a block are not counted).

If the service refuses the set, nothing is saved and the screen shows only *Could not save the bindings.* The service's reasons, which the screen does not show, are:

Service rule	Service message
Every binding has a key	*every binding needs a key*
One binding per key (ignoring case)	*duplicate binding for key '…'*
Source is document, form, process or cross-check	*binding '…': unknown source '…'*
Known **If missing** and **Repeats** values	*binding '…': unknown onMissing '…' / binding '…': unknown cardinality '…'*
A repeating block uses a document source	*binding '…': a repeating block needs source 'document' (got '…')*
A document binding names a role (checklist item, classifier or document type)	*binding '…': a document binding needs a role (checklist item, template or document type)*
Form, process and cross-check bindings name a field	*binding '…': field is required for source '…'*
A key is not both a block's sub-key and bound on its own	*binding '…': already covered by the repeating block '…'*

Every save is recorded in the tenant's audit trail.

Button	What happens	Who	Reversible?
Save	Replaces the template's bindings with the ones on the screen. Takes effect on the next render.	Tenant-management profiles	Yes: change and save again.

Common problems

Message	Cause	What to do
Could not load the bindings for this template.	The template does not exist, was deleted, or the service did not answer.	Check the address; reload.
This template does not have a published version with placeholders yet.	No version is published, or the published one has no placeholders.	Publish a version.
… key(s) without a source: …	Keys of the published version have no binding. They print empty.	Choose a **Source**, or leave them if empty is right.
Bindings for keys that no longer exist in the document (…) — saving removes them.	The published version no longer contains those keys.	Click **Save**.
Could not save the bindings.	The service refused the set (see Save).	Check repeating blocks are on document fields.
A Workflow run stops at Generate document with *needs review*	A **Send for review** binding found no value.	Fix the case data, or change the rule to **Leave blank**.
A document group says no classifiers or no fields	Nothing to bind yet.	Create the classifier and define its fields in Vision.

Related

- Templates — upload and publish the version these keys come from.
- Word add-in — bind placeholders while editing in Word.
- Generate document — the phase that uses bindings.
- Render playground — renders with typed values, not bindings.
- Vision administration overview — classifiers and their fields.

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Render playground

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

Render the published version of a template to PDF in the browser, with values you type, to check the layout before Workflow or your application uses it.

How to get here	**Playground** in the Template menu, or the Dashboard's **Render playground** tile.
Who can use it	Tenant-management profiles can open the screen and list templates. The render itself is authorised by the **API key** you enter, not by your sign-in.
Content status	From source; not observed (`template-fe` `ed4ab73` , `template-be` `cc0f2b0` , 2026-10-22)

The Playground calls the same public render route as your application (see overview — render API). It sends the values you type; the template's field sources are not used. Every render that is not served from the cache counts toward the tenant's monthly render quota and appears in the Dashboard's **Renders (7d)**.

Screen layout

Render playground has three columns.

Pick a template

The templates you can render, each with its name and code. Select one with a click, Enter or Space; it is highlighted. Selecting another template clears the values you typed.

The list shows the tenant's active templates that have a published version — up to 100 of them. A template with only drafts does not appear.

Fill the values

Before you select a template: *Select a template*. If the template has no published version: *Template has no published version*.

Otherwise, one text box per key of the **published** version, labelled with the key, with the key between double braces as its hint. Then the API key, the cache option and the render button.

PDF preview

Before the first render: *Render a template to preview the PDF here*. After a render, the PDF is shown in the column, in an isolated frame. Each new render replaces it.

Fields

Field	What it does	Required	Format / values	Default	Validation messages
One box per key	The value printed in place of that placeholder.	No	Text, printed as typed	Empty	—
API key	A tenant API key for the production environment; sent as the `X-Xtrakt-Api-Key` header. Hidden as you type, and never stored: enter it again after reloading the page.	Yes	The key as issued	Empty	*API key required*
Use cache	Reuse the PDF rendered earlier for the same version and exactly the same values (kept 30 days). A cached answer does not count toward the quota.	No	On / off	Off	—

A box left empty prints as empty.

Buttons and actions

Button	What happens	Who	Reversible?
Render PDF (**…** while rendering)	Renders the published version with the typed values and shows the PDF. Under the button: *Rendered in … ms* and **Cache hit** or **Fresh render**.	Anyone with a valid production API key, from a browser address the environment allows	—

To keep the PDF, use the viewer's own controls in the preview frame (what they offer depends on your browser).

Shared preview link

The Playground has no shared or public preview link. The interface carries text for a shared read-only preview (*Preview shared template*), but no screen of this version shows it.

Common problems

Message	Cause	What to do
API key required	**API key** is empty.	Enter a production API key.
invalid tenant api key	The key is revoked, unknown or not a production key.	Check the key.
render endpoint requires an approved origin or internal service token	The environment does not allow renders from this browser address. Browser renders are off unless the environment allow-lists the address.	Render from your application instead, or ask your administrator whether browser renders can be enabled in this environment.
Another message from the service	The service refused the render — for example the month's quota is used up, the template has no published	Act on the reason; check the template's page and the quota.

Message	Cause	What to do
	version, or the published .docx is larger than 25 MB. The message shown is the service's own reason.	
An error naming status code 429	More than 10 renders a minute with the same key.	Wait a minute.
Pick a template is empty	No active template of the tenant has a published version. If the list could not be loaded, an error message with the service's reason is shown as well.	Publish a version on the template page.
Template has no published version	The template has only drafts.	Publish a version.

Example — from a .docx to a generated PDF

1. **Write the document.** In Word, write a short letter with three placeholders: `nome`, `valor` and `data`, each between double braces (see Placeholder syntax). The Word add-in can insert them for you from the list of fields your processes supply.

```
Prezado(a) {{nome}},
O valor aprovado é {{valor}}, em {{data}}.
```

2. **Create the template. Templates** › **+ New template**, **Name** *Carta de aprovação*, **Save**. Cogneris gives it the code `carta-de-aprovacao` (Create a template).

3. **Upload the .docx.** Drop the file on **Drop a .docx file or click to choose**. *Version 1 uploaded*; **Keys detected (3)** lists `data`, `nome` and `valor` (Upload a version).

4. **Publish it. Publish this version** on **v1**. *Version published* (Versions and publishing).

5. **Render it here. Playground** › select *Carta de aprovação*. Type a name, a value and a date, enter a production **API key**, select **Render PDF**. The PDF appears in **PDF preview**.

6. **For a Workflow process,** bind each key on Field sources — for example `nome` to a field of the identity document, `data` to **Data de hoje** — and put the code `carta-de-`

`aprovacao` in the Generate document phase.

7. **To change the letter,** upload a new version, try it here, and publish it when ready. The code does not change, so the Workflow phase and your application keep working.

Related

- Template administration overview
- Templates
- Field sources
- Word add-in
- Generate document

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Word add-in

> **NOT YET CHECKED ON THE PRODUCT**
>
> This chapter was written from the product's source code and has not yet been checked by a person on a running Cogneris. If what you see on screen differs, the screen is right — please tell your Cogneris contact so the chapter can be corrected.

The Cogneris Word add-in is a task pane inside Microsoft Word. It lets a template author insert the fields a process can actually supply as structured placeholders, convert the placeholders of older templates, and send the open document to Template as a new version — and publish it — without saving a file and uploading it in the web interface.

How to get here	In Word: **Home** tab › **Cogneris** group › **Templates** (**Modelos** in Portuguese). The add-in must first be installed for you (see Install the add-in).
Who can use it	Anyone can open the pane and sign in. Every request it makes to Template (field list, template list, upload, publish) requires one of the roles `customer-admin`, `customer-tech`, `cogneris-admin` (or its older spelling `cogneris-platform-admin`) or `cogneris-tech`, in the tenant of the signed-in account. Without one, each tab shows an error that starts with `403`.
Content status	From source; not observed (`template-addin` `eee52f7`, `template-be` `cc0f2b0`, 2026-10-22)

The add-in follows Word's language. When Word is displayed in Portuguese (any variant), the pane, the sign-in window and the ribbon button are in Brazilian Portuguese; in any other language they are in English. This chapter quotes the English text.

How it fits with the rest of Template

1. Create the template in Template first — the add-in cannot create one. See Templates.
2. Write the document in Word and insert fields from the **Fields** tab.

3. Send it from the **Publish** tab as a new version of that template, and publish it when it is ready.
4. Decide where each placeholder gets its value in the template's bindings. Inserting a field in Word does **not** create a binding. See Bindings.
5. A workflow renders the published version with the Generate document phase.

Install the add-in

The add-in has two separate parts, and both are needed:

Part	Where it lives	What it means for you
The pane (the application itself)	A Cogneris web service at a fixed HTTPS address, for example `word-addin-dev.cogneris.ai` for the development environment.	Word loads the pane from that address **every time** it is opened. Word keeps no copy: if the service cannot be reached, the pane does not open.
The manifest (`manifest.xml`)	Your own Microsoft 365 tenant.	It tells Word that the add-in exists and where to load it from. The service publishes it at `https://` + its host + `/manifest.xml` .

Each environment (development, staging, production) has its own address and its own manifest, with its own add-in id, so two environments can be installed side by side. A non-production manifest can carry a label in the add-in name, for example **Cogneris — Templates (dev)** (**Cogneris — Modelos (dev)** in Portuguese). Ask Cogneris for the address of the environment you use.

For the whole organization (Microsoft 365 admin center)

This is the normal way for a company tenant. It does not go through Microsoft AppSource.

1. In the Microsoft 365 admin center, open **Settings › Integrated apps › Upload custom apps**.
2. Upload the `manifest.xml` file, or give its URL (`https://` + add-in host + `/manifest.xml`).
3. Assign the add-in to the users or groups who author templates.

For a pilot (sideload)

Platform	How
Word for Windows	Put the manifest in a shared network folder. In Word, add that folder under **File › Options › Trust Center › Trust Center Settings › Trusted Add-in Catalogs**. Restart Word, then choose the add-in under **Insert › My Add-ins › Shared Folder**.
Word for Mac	Copy the manifest to `~/Library/Containers/com.microsoft.Word/Data/Documents/wef/`.
Word on the web	**Insert › Add-ins › Upload My Add-in › Browse**, and choose the manifest file.

The Microsoft menu names above are Microsoft's and may vary with your Office version and language.

What the manifest declares

Item	Value	What it means
Name	**Cogneris — Templates**, **Cogneris — Modelos** in Portuguese (plus the environment label, if any)	The name in Word's add-in list.
Description	*Insert process fields into Word templates and publish them to Cogneris.*	Shown in the add-in list.
Office application	Word only.	The add-in does not load in Excel, PowerPoint or Outlook.
Minimum API versions	`WordApi` 1.3 and `DialogApi` 1.1.	Word refuses to install the add-in where these are missing. 1.3 is required because the conversion of old placeholders needs to know whether text is already inside a

Item	Value	What it means
		placeholder; on older Word it would wrap every placeholder twice.
Permission	`ReadWriteDocument`	The add-in reads and changes the content of the open document. It has no access to other documents or files.
Allowed domain	The Cogneris sign-in service of that environment.	Lets the sign-in window open the Cogneris sign-in page.
Ribbon button	**Home** tab, group **Cogneris**, button **Templates** (**Modelos** in Portuguese). Tooltip: *Open the pane to insert process fields and publish this template.*	Opens the task pane.
First-run hint	**Cogneris — Templates**: *Open the Home tab and click Templates to begin.*	Word shows it after installation.

Supported Word versions: Word desktop for Windows or Mac, and Word on the web — any Word that provides the API versions above.

Sign in

The pane does not use your Cogneris browser session: Word runs the pane in its own embedded browser (desktop) or in an isolated frame (web), where the Cogneris session cookie is not available. The add-in therefore signs you in through a separate Office window.

Fields and buttons

Element	What it does
Sign in to your Cogneris account to see the available	Shown until you are signed in.

Element	What it does
fields and publish templates.	
Sign in	Opens a Word dialog window titled **Cogneris — Sign in** on the Cogneris sign-in page. While it is open the button reads **Signing in…** and cannot be clicked again.
Sign-in window	Shows *Signing in…*, moves to the Cogneris sign-in page where you sign in as usual, then shows *Finishing…* and closes itself. The pane then shows its three tabs.
Sign out	At the bottom of the signed-in pane. Forgets the add-in's access token and returns the pane to the sign-in screen.

What is kept. The add-in signs in with the same Cogneris client as the other Cogneris applications. It receives only an access token, which it keeps in memory: never in the browser's storage and never in the document, so the token does not travel with the `.docx` you send to someone. As a result, **closing the pane signs you out**, and you sign in again the next time you open it.

When the session ends. The add-in treats the token as expired one minute before its real expiry. The next action that calls Template opens the sign-in window again instead of failing.

Fields tab

The catalog of what a process can put into the document. Click a field to insert it as a placeholder where your text is selected.

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Search fields…	Filters the list as you type. A field matches when its label or its key	No	Free text	Empty	*No field matches the search.*

Field	What it does	Required	Format / values	Default	Validation messages
	contains the text, ignoring case. A group that is not configured stays visible with its instructions.				
Lock inserted fields against editing	When ticked, each field you insert from now on cannot be edited in Word: nobody can type inside it. It can still be deleted. Fields inserted before you ticked it are not changed.	No	Checkbox	Off	—
Field list	Grouped by origin, each group headed by its name. Each field shows its label, its key, and — when the value may be absent — *Not always filled in — set a default value*.	—	—	—	—

While the list loads, the pane shows *Loading the data contract…*.

Groups and keys

The list comes from Template (`GET /data-contract/fields`). Template builds it for your tenant and keeps it for **5 minutes**, so a classifier or form you just changed can take up to 5 minutes to appear. The group and field names and the instructions below are written by Template in Portuguese and are shown as sent, whatever Word's language.

Group	Comes from	Key format	Always fil
One group per classifier, named after it	The fields that field discovery found for that classifier in Vision.	The classifier name in camel case, without accents and without short connecting words (`de`, `da`, `do`, `e`, `para` …), then a dot and the field path. Example: a classifier "Documento de Identidade — Beneficiário" with a field `cpf` gives `documentoIdentidadeBeneficiario.cpf`. The label is the last part of the path with a capital first letter.	Never — a document is in a case if it uploaded, ar field only if extraction re
One group per Portal form, named after it	The questions of the form's design. Fixed texts (titles, paragraphs) are left out.	The form name in camel case, a dot, and the question's name in the form designer — with accents removed and any character other than letters, digits, `.`, `-` and `_` turned into `_` (a question named *Nome completo* gives `proposta.Nome_completo`). The value still comes from the question itself. The label is the question's label.	Only for que marked requ
Processo (Process)	Fixed.	`process.identifier` (**Identificador do processo**), `process.requester.name` (**Nome do requerente**), `process.today` (**Data de hoje**).	Only `process.t`
Batimento (Cross-check)	Fixed.	`crosscheck.decision` (**Decisão do batimento**), `crosscheck.summary.mismatch` (**Nº de divergências**), `crosscheck.summary.total` (**Nº de verificações**). A placeholder bound earlier to `crosscheck.findingsCount` is read as `crosscheck.summary.mismatch`.	No.

A group with nothing to offer is still shown, with an instruction instead of fields:

Instruction shown (from Template)	Meaning
Ainda não há classificadores neste tenant. Crie um em Classificadores, carregue um exemplar do documento e execute a definição de campos — os campos aparecem aqui logo a seguir.	Your tenant has no classifiers yet. Create one in Vision, upload a sample and run field discovery.
O classificador "…" ainda não tem campos definidos. Abra-o em Classificadores, carregue um exemplar e execute a definição de campos.	That classifier has no discovered fields.
Não foi possível contactar o serviço de classificadores. Tente novamente; se persistir, os campos de documento não podem ser listados agora.	Template could not reach Vision. Document fields cannot be listed now.
Este tenant ainda não tem formulários. Crie um no Portal e desenhe os campos — aparecem aqui logo a seguir.	Your tenant has no Portal forms.
O formulário "…" ainda não tem campos de resposta — só textos fixos. Abra-o no Portal e acrescente perguntas.	That form has only fixed texts.
Não foi possível contactar o serviço de formulários. Tente novamente; se persistir, os campos do formulário não podem ser listados agora.	Template could not reach Portal.

Template reads classifiers and forms with your own sign-in, so you only see what you could see in Vision and Portal yourself. If one form's design cannot be read, that form is left out of the list without a message.

What inserting a field does to the document

A placeholder is two things at once:

a Word content control tagged	cogneris:<key>
whose visible text is	{{<key>}}

When you click a field:

1. Your selection is replaced by the key between double braces, for example {{process.today}}.
2. That text is wrapped in a Word content control (shown with a bounding box) whose tag is `cogneris:` followed by the key and whose title is the field's label.
3. The control can be deleted. It can be edited only if **Lock inserted fields against editing** was off.

Why both: the visible text is what Template detects and replaces when it renders, so the document works exactly like a template written by hand. The content control keeps the placeholder in one piece — Word cannot split it while someone edits the paragraph, and it cannot be half-deleted, which is what happens to typed text.

The add-in only accepts keys made of letters, digits, `.`, `-` and `_` — the same rule Template uses to detect placeholders. A key outside that rule would look correct in Word and then be missing from the keys Template detects.

Inserting a field does not decide where its value comes from. After you send the version, set its bindings in Template: see Bindings. For a field marked *Nem sempre preenchido*, set what happens when the value is missing there.

Buttons and actions

Button	What happens	Who	Reversible?
A field in the list	Replaces the selected text with the placeholder, as described above.	Signed-in user	Yes — Word's undo, or **Remove** in the **Document** tab.

Document tab

What is in the open document right now, and the conversion of placeholders typed as plain text. The tab reads the document when you open it.

Fields

Element	What it shows
Markers to convert (…)	Only when the document has placeholders typed as plain text — in the body, in the headers and footers of any section (first page, even pages and the others), and in footnotes and endnotes where Word supports

Element	What it shows
	`WordApi` 1.5. Placeholders with spaces inside the braces (`{{ nome }}`) are included. The number is how many different keys. Below it: *These {{campo}} are loose text. Once converted they stop breaking when someone edits the document — the visible text does not change, so the generated PDF stays exactly the same.* Then one line per key: the key between double braces, `×`, and how many times it appears.
Fields in the document (…)	The number of Cogneris placeholders (content controls whose tag starts with `cogneris:`). Content controls of other origins are ignored. With none: *No fields yet. Use the Fields tab to insert the first one.*
One row per placeholder	Its key, followed by · *locked* when it cannot be edited. When its visible text is no longer exactly the key between double braces (someone typed inside it), the row warns: *Text changed (…) — it will not be replaced*.

Buttons and actions

Button	What happens	Who	Reversible?
Refresh	Reads the document again. While reading it shows **Reading the document…** and the other buttons wait.	Signed-in user	—
Lock converted fields against editing	The same option as **Lock inserted fields against editing** in **Fields** — ticking one ticks the other. When ticked, **Convert all** locks every placeholder it creates.	—	—
Convert all	Wraps every plain-text placeholder found (body, headers, footers and, where supported, notes) in a Cogneris content control (tag `cogneris:` + key, title = key, bounding box, can be deleted, locked only when the lock option is ticked), then reads the document again. A placeholder already written as the key between double braces	Signed-in user	Yes — Word's undo, or **Remove** on each placeholder.

Button	What happens	Who	Reversible?
	keeps its visible text. One with spaces inside the braces is rewritten without them (`{{ nome }}` → `{{nome}}`), because Template's renderer fills only the exact form. Running it twice does nothing the second time: text already inside a Cogneris placeholder is skipped.		
Locate	Scrolls the document to that placeholder and selects it.	Signed-in user	—
Remove	Asks for confirmation on the row itself: the button becomes **Confirm removal**, next to **Cancel**. Confirming removes the content control but **keeps its text**, so the key between double braces stays in the document as plain text (and appears again under **Markers to convert**). Delete the text yourself if you no longer want the placeholder. On a locked placeholder the button is disabled: unlock it first in the content control's properties in Word.	Signed-in user	Yes — Word's undo, or **Convert all**.

Converting an older template

Every template written before the add-in has its placeholders as plain text. To convert one:

1. Open the template's `.docx` in Word. Template's web interface and service do not offer a download of an uploaded version (`template-be` `cc0f2b0`), so use your own copy of the file that was uploaded.
2. Open **Document** and click **Convert all**.
3. Send it as a new version from the **Publish** tab.

A placeholder already written as the key between double braces keeps its visible text, so the rendered PDF stays the same; only the structure of the document improves. A placeholder with spaces inside the braces is rewritten without them, so that Template's renderer fills it.

What conversion does not find: on a Word without `WordApi` 1.5, placeholders in footnotes and endnotes are not scanned, although Template detects them.

Publish tab

Sends the open document to Template as a new draft version of an existing template, then — as a second, separate click — publishes it.

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Target template	The template that receives the new version. Lists the templates of your tenant by name, most recently changed first, without archived ones. The list is limited to the **20** most recently changed templates and has no search: to send to an older template, change it in Template first so it moves up, or upload in the web interface.	Yes	List; first entry **Choose…**, not selectable	**Choose…**	The send button stays disabled until a template is chosen.
Notes for this version	Free text stored with the version. Hint: **What changed**.	No	Free text	Empty	—

Buttons and actions

Button	What happens	Who	Reversible?
Upload as a new version	Reads the open document from Word as a `.docx` (in slices of up to 4 MB) and sends it to Template (`POST /templates/…/versions`). The file name is the document's own name, or `documento.docx` when Word gives none. While sending it reads **Uploading…**. Template checks the file, stores it, creates a **draft** version with the next number and detects its placeholder keys in the body, headers, footers, footnotes and endnotes. If the content is identical to an earlier version of the same template, the stored file is reused but a new version is still created. The upload is recorded in the audit log. Then the pane shows *Version … created* with the version number, one line per detected key, and the **Publish this version** button. A key that matches no field of the data contract is marked *No source in the data contract*. When the document is identical to a version already sent: *This document is identical to a version already uploaded.* With no keys: *No fields found in the document.*	Roles listed in the table at the top	A draft is not used by renders. Template has no way to delete a version from this pane.
Publish this version	Appears only after a successful send, for the version just sent. Publishes it (`POST /templates/…/versions/…/publish`): the version is marked published, with the date and your e-mail, and becomes the template's published version — **from now on every render of this template in the whole tenant, including the Generate document phase of every workflow, uses it.** The publication is recorded in the audit log. Then the pane shows *Published. This is the version the render now serves.*	Roles listed in the table at the top	Yes — publish another version (see Templates).

Publishing is deliberately a separate click: it changes a production document for the whole tenant, and must never be the side effect of an upload. To publish later, send now and publish from Template's web interface.

What Template checks on upload

Check	Message (from Template)
A file is present and not empty.	*file is required*, *file is empty*
Size at most 50 MB.	*file exceeds the maximum allowed size of 50 MB*
`.docx` only, and a real Word document.	*only .docx files are allowed*
The archive is not abnormally large or compressed (protection against "zip bombs").	Decided by the service.
The template exists in your tenant.	*template not found*
For publish: the version belongs to the template.	*version not found for this template*

Several messages from Template are shown together, separated by `;`.

Common problems

Message	Cause	What to do
Word shows its own "add-in could not start" error, or the pane stays blank.	Word could not load the pane from the Cogneris add-in service (service unreachable, or blocked by your network). Word keeps no copy of it.	Check that Word can reach the add-in host. If it persists, contact Cogneris support.

Message	Cause	What to do
This pane only works inside Microsoft Word.	The pane was opened outside Word (for example in a browser or another Office application).	Open it from Word: **Home › Cogneris › Templates**.
The sign-in window opens, goes white and closes, with no message.	The environment's manifest does not allow the Cogneris sign-in domain.	Contact Cogneris support; this is fixed in the manifest, not in Word.
Could not open the sign-in window., or a message from Office.	Word refused to open the sign-in dialog.	Try again; check that Word does not block pop-up dialogs for add-ins.
The sign-in window was closed before finishing.	The sign-in window was closed before signing in finished.	Click **Sign in** again and complete the sign-in.
Sign-in cancelled. or a message from the sign-in service	Sign-in was refused or cancelled; the sign-in service's own description is shown when it gives one.	Sign in again with a valid Cogneris account.
The sign-in session expired. Try again.	The sign-in window lost its state during the round trip.	Click **Sign in** again.
Invalid sign-in response (state).	The answer from the sign-in service did not match the request.	Click **Sign in** again. If it repeats, contact Cogneris support.
Unexpected response from the sign-in window.	The sign-in window returned something the pane cannot read.	Try again; contact Cogneris support if it repeats.

Message	Cause	What to do
A message starting with *The sign-in service answered*	The sign-in service refused to issue the token.	Try again; contact Cogneris support with the message.
Could not load /config.json (…). Check CONFIG_JSON on Cloud Run. or *config.json does not define ssoAuthority — sign-in cannot start.*	The add-in service of that environment is misconfigured.	Contact Cogneris support.
A message starting with `403` on any tab	Your account has none of the roles Template requires (see the table at the top), or the add-in is installed for an environment where your account is not a member.	Ask your tenant administrator for the `customer-admin` or `customer-tech` role.
No response from the server. Check your connection.	Template did not answer, or the request took more than 60 seconds.	Check the connection and try again. A large document on a slow connection may take long to send.
Empty response from the server. or *The server returned an error.*	Template answered without content, or with an error it did not describe.	Try again; contact Cogneris support if it repeats.
A group of **Fields** shows an instruction instead of fields.	That source is not configured, or could not be reached.	Follow the instruction (see Groups and keys).
"…" is not a valid field key. Use unaccented letters, digits, dot, hyphen or underscore.	The key has characters Template cannot detect. Keys offered in **Fields** are already valid: Template turns spaces and accents in form question	Use letters, digits, `.`, `-` or `_`.

Message	Cause	What to do
	names into `_` and plain letters.	
The field "..." is locked. Unlock it in the Word content control properties before removing it.	You tried to remove a locked placeholder.	Unlock it in the content control's properties in Word, then **Remove**.
Text changed (...) — it will not be replaced	Someone typed inside the placeholder, so its visible text is no longer the key between double braces. Template will neither detect nor fill it.	Click **Remove**, delete the text, and insert the field again from **Fields**. Tick **Lock inserted fields against editing** to prevent it.
Word did not hand over the document. or a message starting with *Could not read block*	Word could not give the add-in a copy of the open document.	Save the document, close and reopen the pane, and send again.
file exceeds the maximum allowed size of 50 MB	The document is larger than Template accepts.	Reduce the document (for example compress images) and send again.
The template is missing from **Target template**.	It is archived, or it is not among the 100 most recently changed templates.	Upload in the web interface. See Templates.
A rendered document has a blank where a field should be.	The field has no binding, or its value was missing in the case.	Set the binding and what happens when the value is missing: see Bindings.

Known limitations in this version

- *No source in the data contract* compares each key with the data contract only. A key you bound by hand on Bindings is still marked if no contract field has that key.

- Only English and Brazilian Portuguese: Word in any other language shows the English text.

Related

- Templates — create templates, see their versions and publish from the web interface.
- Bindings — where each placeholder gets its value, and what happens when it is missing.
- Generate document — the Workflow phase that renders the published version.

Last reviewed 2026-10-22 · Version 1.0.0 · Owner: product-documentation

Document Center administration overview

For company administrator and technical profiles. What the Document Center is, and what every other profile can do in it (Dashboard, upload, opening a shared link), is described in the user manual's *Document Center* part.

> **CONTENT STATUS**
>
> This part was written from the interface and service source of the build serving DEV on 2026-10-09 (Document Center interface `f3b92b7`, service `bb1cb28`). On 2026-10-17 and 2026-10-18 every screen was observed in DEV with a synthetic company administrator (interface `7c2a77e` / service `cfb72ae`, then interface `44d2f38` / service `6fd4193`), after the fixes of process status (XTRAK-2548), search (XTRAK-2549) and **Share** (XTRAK-2499); each screen of Document Center administration screens states its own content status, and the items still to be observed are listed under *Not observed* there.

What you manage here

Behind **My Drive** the Document Center keeps three kinds of content:

- **Processes** — cases that arrived with their documents, grouped by **process type** (the form or workflow they came from). A process carries the requester, an identifier, the extracted data, the checklist of expected documents, the documents generated by the workflow, and the history of every review decision.
- **The Inbox** — loose documents (uploaded, received by e-mail or through the API) waiting to be linked to a process, archived or discarded.
- **Folders** — the folder tree (**Explorer**) that holds documents, including loose and archived ones. From a document you can create a share link.

Processes are **read-only** in the Document Center: their documents are reviewed and decided in Workflow, and the Document Center shows the outcome and the trail.

Who may do it

Screen	Profiles
Dashboard	Every profile that includes the Document Center
My Drive, process types, **Processes**, process detail	Company administrator and technical profiles
Inbox, **Explorer**, document detail and share links, **Search**	Company administrator and technical profiles

The platform menu shows **My Drive** only to administrator and technical profiles. These screens give access to your company's processes and documents, so assign those profiles only to people who need that access. Profiles are assigned in *Manage members and profiles*.

Journeys in this part

1. Triage the Inbox — link a loose document to its process, archive it or discard it.
2. Follow a process — find a process from My Drive and read its documents, data and decisions.
3. Organize and share documents — create folders, upload into them, open a document and share it by link.
4. Search documents — full-text search with dates in plain words.

Screens in this part

The Document Center administration screens chapter describes **My Drive**, the process type page, **Processes**, **Process details**, **Inbox** and its **Link to a process** dialog, **Explorer**, **Document details** and **Search documents**.

Known limitations

- Share links are listed and revoked on the document's page, under **Active share links**.
- There is no action to delete, move, rename or tag a document from the interface; only Inbox documents can be discarded.
- The **Integrations** and **Comments** views of a process are placeholders.
- The Inbox shows the 50 most recent loose documents.
- Pinned process types are remembered in your browser only.

Last reviewed 2026-10-18 · Version 1.0.0 · Owner: product-documentation

Triage the Inbox

Use this journey to clear loose documents — files that arrived by e-mail, upload or API without a process.

Before you start

- You have a company administrator or technical profile, and **My Drive** appears in the Document Center menu.
- The **Inbox** card in **My Drive** shows a number greater than zero. A card with documents waiting is highlighted.

Steps

1. In the Document Center menu, select **My Drive**, then the **Inbox** card. The **Inbox** lists one row per document: its origin (**Email**, **Upload** or **API**), name, and **Received** date and time.
2. For each document, choose one action:
   - **Link** — the document belongs to a process. Go to step 3.
   - **Archive** — keep the document without a process. It leaves the Inbox at once.
   - **Discard** — the document is not needed. A confirmation asks *Discard document? — This permanently removes the document.* Select **Discard** to confirm, or **Cancel**.
3. In **Link to a process**:
   - Under **Suggested**, the Document Center lists processes that match identifiers or e-mail addresses read from the document, with the reason (**by ID** or **by email**). If there is none, it shows *No automatic suggestion*.
   - Under **Search process**, type a name, requester or identifier and select **Search process**.
   - Select **Link** on the right process. The button shows **Linking…** and the dialog closes.
4. *(Optional)* To add a loose file yourself, select **Upload loose file** at the top of the Inbox and pick the file. It appears in the list with origin **Upload**.

Expected result

- A **linked** document leaves the Inbox and becomes one of the process's documents.
- An **archived** document leaves the Inbox, gets the status *Archived* and stays in its folder; it can be found through **My Drive** › **Archived / Loose files**.

- A **discarded** document is deleted. This cannot be undone.

How to confirm it worked

- The row is gone from the Inbox, and the **Inbox** card count in **My Drive** went down.
- For a linked document: open the process (see Follow a process) and find the document among its thumbnails.
- For an archived document: its timeline shows *Archived*.

If something goes wrong

What you see	What it means	What to do
The row comes back after an action	The action failed and the list was reloaded	Try again. If it repeats, contact support.
Could not load the inbox with **Retry**	The list did not load	Select **Retry**.
Link in the dialog does nothing visible	The link failed; the dialog stays open	Try again or choose **Cancel**.
No process found	The search matched no process	Search by requester name or identifier exactly as received.
Nothing to triage	The Inbox is empty	Nothing to do.

Observed on 2026-10-17 in DEV: uploading a loose file and opening **Link to a process**; no document was archived, discarded or linked.

Not observed: which identifiers the suggestions read from a document, and the Inbox with more than 50 documents (only the 50 most recent are listed).

Last reviewed 2026-10-17 · Version 1.0.0 · Owner: product-documentation

Follow a process

Use this journey to find a process and read everything the Document Center keeps about it: its documents, the data extracted from them, the checklist, the review decisions and the audit trail.

Before you start

- You have a company administrator or technical profile, and **My Drive** appears in the Document Center menu.
- The process has arrived — for example a Portal submission that ran through Workflow. Processes appear on their own; you do not create them here.

Steps

Find the process

1. In the Document Center menu, select **My Drive**. One card per **process type** shows how many processes it has and, when relevant, *n awaiting review* or *n in progress*. The same types are listed under **My Drive** in the side menu.
2. Select the card of the process type. The page shows the **Recent processes** of that type (up to 10), each with its status, requester, identifier, age (*just now*, *5h*, *3d*), number of documents and checklist progress.
3. Select the process. If it is not among the recent ones, select **View all**: the **Processes** list opens filtered by that type.
4. In **Processes**, narrow the list if needed: type in *Search requester or identifier...*, choose a status in **All statuses**, and select **Filter**. Use **Previous** and **Next** to page (25 per page). Select a row to open the process.

To keep a type at the top of **My Drive**, select **Pin** on its card.

Read the process

5. **Process details** opens. The header shows the process name, its status and, when the process has a checklist, a ring with the documents received out of those expected. Below it: **Requester**, **Identifier**, **Workflow**, **Received** and **Source**.
6. On the left, select a document thumbnail to show it in the preview.

7. On the right, use the icon bar (hover an icon to see its name):
    - **Extracted data** — the fields read from the selected document, under its document type.
    - **Checklist** — each expected document, ticked when received, or marked **Pending** or **Optional**; then the **Form answers**.
    - **Generated** — documents the workflow produced; **Open in the viewer** or **Download**.
    - **Timeline** — the **Correction history** (one round per return to the requester) and every event of the process.
    - **Cross-check** — what agreed across the documents and what did not. It never rejects the case; you decide.
    - **Workflow** — the review outcome (**Approved**, **Refused** or **Returned to requester**), who decided and when, then each phase of the run and its status.
    - **Audit** — the **Human decisions** and the **Immutable log** of events.
    - **Versions** — every version of every document, with **Download**.

Expected result

You can answer, for one process: which documents arrived and which are missing, what was read from them, who decided what and when, and whether the requester was asked to correct something.

How to confirm it worked

The status in the header matches the outcome in **Workflow**: for example *Awaiting requester correction* with the outcome **Returned to requester**, and a round in **Timeline** › **Correction history**.

Process statuses

Status	Meaning
New	Received; its workflow has not started
Processing	The workflow is running — including a reprocessing after the requester resent documents
Awaiting review	Waiting for a reviewer in Workflow

Status	Meaning
Awaiting requester correction	A reviewer refused one or more documents and the requester was asked to correct and resend
Completed	Finished
Rejected	A reviewer refused documents and there is no requester to return them to (the process did not come from a Portal link)
Failed	The workflow run stopped and will not resume on its own; someone has to look at it. This status has no entry in the **All statuses** filter.

The **Dashboard** counts *New*, *Processing* and *Awaiting review* as **In progress**.

If something goes wrong

What you see	What it means	What to do
Could not load this process.	The process did not load	Reload the page.
Preview unavailable	The selected document has no stored file, or its preview did not load	Select another thumbnail, or download it from **Versions**.
Workflow shows *No workflow run linked to this process yet.*	The process did not come from a workflow, or the run is not linked	Nothing to fix here; check the process **Source**.
Audit › **Human decisions** says some decisions are from before the trail started	The process is older than the decision trail	Earlier decisions are not in the trail; the **Immutable log** still lists the events it has.
A type card is missing from **My Drive**	No process of that type has arrived	Nothing to fix here.

The **Integrations** and **Comments** icons open a placeholder: *This view arrives in a later phase.*

Observed on 2026-10-18 in DEV: **My Drive**, a process type page, **Processes** and every view of **Process details** for a process whose workflow run finished; the status badge showed *Completed*.

Not observed: how a **Failed** process is shown.

Last reviewed 2026-10-18 · Version 1.0.0 · Owner: product-documentation

Organize and share documents

Use this journey to create folders, upload documents into them, open a document and give someone outside your company a link to it.

Before you start

- You have a company administrator or technical profile, and **My Drive** appears in the Document Center menu.
- To share: the person you share with needs no account. Anyone who has the link can open the document, so send it only to the intended person.

Steps

Create a folder and upload into it

1. In the Document Center menu, select **My Drive**, then **Archived / Loose files**. (From the **Dashboard**, **Browse folders** opens the same screen.) The **Explorer** opens: the folder tree on the left, starting at **/ (root)**, and the documents of the selected folder on the right.
2. To create a folder, select where it goes — **/ (root)** or a folder — type its name in **Folder name** and select **Create**. Reload the page to see the new folder in the tree.
3. Select a folder in the tree. Use ▸ to open a folder and see its subfolders.
4. Drop files on **Drop files here or click to choose**, or select it and pick files. You can upload several at once. Each file appears in the list as a new document.

Until a folder is selected, the drop area is disabled and reads *Select a folder before uploading files.*

Open a document

5. Select a document in the folder (or in search results). **Document details** opens: the name, then *Status: … · Confidentiality: …*, a preview of the current version, and two tabs.
6. In **Details** you find:
    - the status and confidentiality badges;

- the **AI panel**, when the document was classified: **Document type**, **Confidence** (a percentage) and **Fraud** (*Possible fraud* or *No fraud detected*). The **Fraud** badge only says whether the document was blocked by a fraud policy. *No fraud detected* does **not** mean a fraud check ran or found the document clean: today it appears on every document that came through a workflow extraction, checked or not. The graded result of the opt-in fraud check (*Trusted*, *Review recommended*, *Suspicious*, with its signals) is shown in the workflow's **Fraud review** queue and in Vision — see Fraud Detection;
- **Versions** — each version with its number, file name and size, and ✓ *signed* when it carries a signature;
- **Tags**.

7. In **Timeline**, every event of the document is listed: created, versions uploaded, viewed, downloaded, moved, shared, status changes and others.

Share it

8. In **Document details**, select **Share**. A banner shows **Link copied to clipboard** and the link. Select **Copy again** if needed, or **Dismiss**. The banner closes by itself after a few seconds.
9. Send the link to the person by your own channel (e-mail, chat). What they see is described in the user manual, *Open a shared document*.

Expected result

- The folder exists and holds the uploaded documents.
- The shared link opens the document's current version for whoever has it, with download allowed, without password, expiry or access limit.
- The document's timeline shows *Shared* when the link is created, and *Downloaded* each time the link is opened.

How to confirm it worked

- The uploaded documents are listed in the folder.
- In **Document details** › **Timeline**, a *Shared* event appears after you select **Share**.

If something goes wrong

What you see	What it means	What to do
The new folder is not in the tree	The tree is loaded when the page opens	Reload the page.
We couldn't load documents for this folder. with **Try again**	The list did not load	Select **Try again**.
Preview not supported for this file type.	Only PDF, PNG and JPG are previewed	**Document details** has no download button. If the document belongs to a process, download it from the process **Versions** view.
A link was shared by mistake	The link keeps opening until it expires or is revoked	On the document's page, under **Active share links**, select **Revoke** on that link. People who already opened it lose access too.

Known limitations

- There is no action to move, rename, tag, change the status or delete a document from the interface.
- A revoked link cannot be restored: create a new one.

Observed in DEV: creating a folder, uploading a PDF and opening **Document details** with its preview, versions, tags and **Timeline** (2026-10-17); **Share** › **Create share link** › **Create link**, and the link opened without an account (2026-10-18). The dialog's text suggests a password, an expiry date or an access limit, which it does not offer (XTRAK-2592).

Not observed: an upload failure in the **Explorer** (the source shows no error message for it).

Last reviewed 2026-10-22 · Version 1.0.1 · Owner: product-documentation

Search documents

Use this journey to find a document by its name, its content, or a phrase that includes a date.

Before you start

You have a company administrator or technical profile. Search covers the text extracted from documents when they were uploaded.

Steps

1. Type in the search box at the top of the Document Center (*Search processes and documents*) and press **Enter**. From the **Dashboard**, **Search documents** opens the same page. **Search documents** opens and runs the search.

2. To search again, type in the box on the page (*Try: RG sent today, contracts this month…*) and select **Search**.

3. To limit by date, write the period in plain words with the rest of the search. The page shows how it read your phrase under **Understood as:**, for example `"contracts"` and **This month**.

You write (English or Portuguese)	Period
today / *hoje*	Today
yesterday / *ontem*	Yesterday
this week / *esta semana*	From Monday to now
this month / *este mês*	This month
this year / *este ano*	This year
last 30 days / *últimos 30 dias*	The last N days (up to 999)

4. *(Optional)* Under **Filters**, select a value to narrow the results; select it again to remove it.

5. Select a result to open its **Document details**.

Expected result

Each result shows the document name, its file type (for example **PDF**), the matching passage with the words highlighted, its date and its relevance score.

How to confirm it worked

The document you expected is in the list, and the **Understood as:** line shows the text and period you meant.

If something goes wrong

What you see	What it means	What to do
No results.	Nothing matched the text and period	Remove the period or a filter, or use fewer words.
Understood as: shows no period	The date phrase was not recognized	Use one of the phrases in the table above.

Known limitations

- The box at the top says *processes and documents*, but the results are documents only. To find a process, use **Processes** (see Follow a process).

Observed on 2026-10-18 in DEV: a word in the names of synthetic documents returned them with **Understood as:**, the filters **MIME**, **Status**, **Confidentiality** and tag keys, and a document uploaded minutes before.

Not observed: how the relevance score is computed, and whether text inside scanned images is searchable.

Last reviewed 2026-10-18 · Version 1.0.0 · Owner: product-documentation

Document Center administration screens

Detailed reference for the Document Center screens managed by company administrator and technical profiles. Labels are quoted as they appear in the English interface. Content was read from the source of the build serving DEV on 2026-10-09 (Document Center interface `f3b92b7`, service `bb1cb28`); on 2026-10-17 the *Inbox*, the *Link to a process* dialog, the *Explorer* and *Document details* were observed in DEV with a synthetic company administrator (Document Center interface `7c2a77e`, service `cfb72ae`), and on 2026-10-18 *My Drive*, the process type page, *Processes*, *Process details*, *Search documents* and the **Share** button (Document Center interface `44d2f38`, service `6fd4193`). Every screen below says **Observed**. On 2026-10-22 the fields, actions and messages of every screen below were checked against the Document Center interface source (`a8e9f24`) and service source (`d3ab083`); those details are marked as not observed in each screen's *Content status*. On 2026-10-23 the *Processes*, *Process details* (**Versions**), *Inbox*, *Link to a process*, *Explorer*, *Document details* and *Search documents* screens were updated to the behaviour of Document Center interface pull request #143 and service pull request #171 (WOR-168); not observed.

Screenshots are not included in this edition. The **Dashboard** and the **Upload document** dialog are described in the user manual's *Document Center screens*.

Who can use these screens. The service answers the screens below only for a company administrator or technical profile (and the Cogneris administrator and technical operators). Another profile that opens them by address gets empty or error states, whatever the side menu shows. There is no separate audit screen and no "shared with me" screen: the audit record lives in the **Audit** and **Timeline** views of Process details and in the **Timeline** of Document details.

The top bar. Every screen below has a search box, *Search processes and documents*. Pressing **Enter** opens Search documents and runs the search; an empty box opens the page without searching. The breadcrumb starts at **ECM**, which opens the Dashboard.

My Drive

How to get here	Menu **My Drive**
Who sees it	Company administrator and technical profiles
Content status	Observed (2026-10-18; on 2026-10-17 a profile without administrator access that opened it by address got *This page cannot be opened*, although the side menu still showed **My Drive**). Field details checked against the Document Center interface source (`a8e9f24`) on 2026-10-22; not observed.

Header: **My Drive** — *Your processes and documents in one place*.

Card	Content	Opens
Inbox	*Documents awaiting triage* and the count; highlighted when not empty	**Inbox**
One per process type	Type name and its number of processes (a bare number in the corner), and *… awaiting review* or, when none awaits review, *… in progress*; **Pin** / **Unpin**	Process type page
Archived / Loose files	*Documents without a process*	**Explorer**

A process type groups the processes captured from the same form; processes without a form are grouped by workflow, then by source. Pinned types come first; otherwise the most active types come first. Pins are kept in your browser only. The side menu lists the same process types under **My Drive**.

Buttons and actions

Button	What happens	Who can use it	Reversible?
Inbox card	Opens the Inbox	Company administrator,	—

Button	What happens	Who can use it	Reversible?
		technical profile	
A process type card	Opens the process type page	As above	—
Pin / **Unpin** (star on a type card)	Moves the type to the top, or back. Kept in this browser only; another browser or a cleared browser storage loses it.	As above	Yes
Archived / Loose files	Opens the Explorer	As above	—
Retry	Loads the drive again	As above	—

Common problems

Message	Cause	What to do
Loading your drive…	The drive is loading	Wait
No processes yet — Processes captured from your forms will appear here	No process exists yet. The **Inbox** and **Archived / Loose files** cards still show.	Nothing to fix
Could not load your drive with **Retry**	The process types could not be loaded	Select **Retry**; if it persists, contact support
The **Inbox** card shows **0** although documents wait	The Inbox count could not be loaded; the screen shows 0 instead of an error	Open the **Inbox** to check

Process type page

How to get here	**My Drive** › a process type card, or the type under **My Drive** in the side menu
Content status	Observed (2026-10-18, card selected and the page opened by its own address). Field details checked against the Document Center interface source (`a8e9f24`) on 2026-10-22; not observed.

Header: the type name, *Recent processes*, **View all** (opens **Processes** filtered by the type) and **Back to My Drive**. Up to 10 process cards, newest first.

Card part	Content
Top	Status icon and status badge
Name	The process name
Chips	Requester and identifier, each only when known
Age	*just now* (under an hour), *…h* (hours) or *…d* (days) since the process arrived; hover for *Received* and the date. For an open process (*New*, *Processing*, *Awaiting review*) the age is tinted from 2 days and more strongly from 5 days.
Progress	The checklist ring (received / expected) when the process has a checklist; otherwise the number of documents, *… docs*

Buttons and actions

Button	What happens	Who can use it	Reversible?
Back to My Drive	Opens My Drive	Company administrator, technical profile	—
View all	Opens Processes filtered by this type	As above	—

Button	What happens	Who can use it	Reversible?
A process card	Opens Process details	As above	—
Retry	Loads the page again	As above	—

Common problems

Message	Cause	What to do
No processes of this type yet	The type has no process, or the address names a type that does not exist	Go **Back to My Drive** and choose a type
Could not load your drive with **Retry**	The processes could not be loaded	Select **Retry**

Processes

How to get here	**View all** on a process type page (filtered by the type), or **Processes** in the breadcrumb of *Process details* (all processes)
Content status	Observed (2026-10-18, list and status filter). Field details checked against the Document Center interface source (`a8e9f24`) and service source (`d3ab083`) on 2026-10-22; not observed.

Header: **Processes** — *Business processes captured from the portal and workflow.*

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Search requester	Shows processes whose name,	No	Free text	Empty	—

Field	What it does	Required	Format / values	Default	Validation messages
or identifier…	requester name, requester e-mail or identifier contains the text, ignoring case. Applied with **Filter** or **Enter**.				
All statuses	Shows processes in one status. Applied as soon as a status is chosen.	No	**All statuses**, *New*, *Processing*, *Awaiting review*, *Awaiting requester correction*, *Completed*, *Rejected*	**All statuses**	—

A type filter coming from **View all** shows under the filters as *Type: …*, with **Show all types** to clear it.

Columns

Column	Content
Flow	Status icon; form or workflow name (or the source: Portal, Workflow, API, ECM) and received date
Subject	Process name
Requester	Name and e-mail; **—** when unknown

Column	Content
Identifier	Process identifier; **—** when unknown
Docs	Number of documents
Status	Status badge

Buttons and actions

Button	What happens	Who can use it	Reversible?
Filter	Applies the text filter and goes back to page 1	Company administrator, technical profile	Yes
Show all types	Clears the type filter that came from **View all** and lists every process	As above	Yes
Previous / **Next**, *Page … of …*	25 processes per page	As above	—
A row	Opens Process details	As above	—
Try again	Loads the list again	As above	—

Common problems

Message	Cause	What to do
No processes yet.	No process matches the filters, or none exists	Clear the text and choose **All statuses**
Could not load processes. with **Try again**	The list could not be loaded	Select **Try again**

Process details

How to get here	A process card or a row in **Processes**
Content status	Observed (2026-10-18, a *Completed* process from a Portal form, every view opened, also by the page's own address). Field details checked against the Document Center interface source (`a8e9f24`) on 2026-10-22; not observed.
Changes data?	No. Every view is read-only.

Area	Content
Header	Process name, checklist ring (received / expected, shown when the process has a checklist; green only for a *Completed* process), status
Summary	**Requester** (name and e-mail), **Identifier**, **Workflow**, **Received** (date and time), **Source** (as stored, for example `portal`); **—** when unknown
Left	Document thumbnails, each titled with its document type, and the preview of the selected document (*Preview unavailable* when it cannot be shown). The first document that has a file is selected when the page opens.
Right	Icon bar; hover an icon for its name

View	Content	Empty / error
Extracted data	Fields read from the selected document, under its document type, sorted by name. Field names are made readable from the stored key; yes/no values show as ✓ / ✗; words in capitals are shown in normal case.	*No extracted data yet for this process.* (the selected document has none)

View	Content	Empty / error
Checklist	Expected documents with ✓ received, or ○ and **Pending** or **Optional**, under *Checklist · …/…*; for processes from before checklists were recorded, the received documents only; **Form answers** as name and value	*This process has no checklist.*
Generated	Documents produced by the workflow: **Open in the viewer** (the opened one is marked *showing*), **Download**	*This process has not produced any documents yet.*; *Download failed*
Timeline	**Correction history** and the process events (see below)	*No correction cycles recorded.*; *No events yet.*
Cross-check	*What agreed across the documents of this case, and what did not. Nothing here rejects the case — you decide.*; **All agreed** or **Needs a look**; per check **Match**, **Divergence**, **Inconclusive** or **Not applicable**, the check's name, the two values compared with the document and field each came from, and the reason in the interface language (for example *The dates differ.* or *One of the documents this check needs is not in the case.*; a sentence written by AI shows as written, and a reason this build does not know shows as stored); **AI** marks checks made by AI	*No cross-check ran on this case.*
Workflow	Outcome (**Approved**, **Refused**, **Returned to requester**), who and when, *… approved · … refused*; workflow name and status; phases (**Classification**, **Extraction**, **Crops**, **Split**, **Human review**, **Delivery**, **Condition**) with status and error	*No workflow run linked to this process yet.*

View	Content	Empty / error
Audit	**Human decisions**; **Immutable log** (**Event**, **When**, **Actor**, **IP**, **Document**, **Hash**) — *Immutable trail — every event is hash-chained, so tampering is detectable.*	*No decision has been recorded for this case.*; *No audit events yet.*
Versions	Each document's versions, **Current**, **Signed**, **Download**	*No documents in this process.*
Integrations, **Comments**	Placeholder	*This view arrives in a later phase.*

When a reviewer reserved a document without deciding, the decision trail shows *reserved by …*. When the process has decisions from before the trail started, it says so and those decisions are missing.

Timeline details. Correction history lists each round (**Round** and its number) with the time it was **Returned to requester**, the **Actor**, the **Reasons** (document and reason), the **Notification attempt** time, the **Channel**, the **Notification outcome** (*Sent*, *Suppressed*, *No channel*), and when the **Requester resent documents** and **Reprocessing started**; **—** when not recorded. The events list shows up to 100 events, each with its name, date and time, the user's identifier and *doc* plus the first 8 characters of the document's identifier.

Audit details. Human decisions lists each decision (**Approved**, **Refused**, **Returned to requester**, *Requester resent*) with the document, the queue, the reviewer (*no reviewer recorded* when unknown), the time, the reviewer's note and *before the trail began* where it applies. In the **Immutable log**, **Actor** is the user's identifier, **IP** shows the browser details on hover, **Document** is the first 8 characters of the document's identifier and **Hash** the first 12 characters (the full hash on hover).

Buttons and actions

Button	What happens	Who can use it	Reversible?
A thumbnail	Shows that document in the preview and in **Extracted data**	Company administrator, technical profile	—
‹, **›**, *Page … of …*	Turn the pages of a PDF in the preview	As above	—
An icon in the icon bar	Opens that view. **Cross-check**, **Workflow** and **Versions** load the first time they are opened.	As above	—
Open in the viewer (**Generated**)	Shows the generated document in the preview	As above	—
Download (**Generated**)	Downloads the generated document; reads **Downloading…** meanwhile	As above	—
Download (**Versions**)	Opens that version in a new browser tab; reads **Preparing…** meanwhile	As above	—

Common problems

Message	Cause	What to do
Could not load this process.	The process does not exist or could not be loaded. There is no retry button.	Reload the page or go back to **Processes**
No documents.	The process has no document	Nothing to fix
Preview unavailable	The selected document has no file, or the preview could not be	Select another document, or

Message	Cause	What to do
	prepared	download it from **Versions**
Could not load the workflow.	The workflow could not be loaded	Open another view and come back to **Workflow**
Could not load the cross-check.	The cross-check could not be loaded	Open another view and come back to **Cross-check**
Could not load the timeline.	The events could not be loaded; **Correction history** and the **Immutable log** are affected too (*Could not load the audit trail.*)	Reload the page
Correction history is unavailable.	The service did not return the correction rounds	Reload the page
Could not load the decision trail.	The decisions could not be loaded	Reload the page
Could not load versions.; *No versions.* for a document	The versions could not be loaded; or that document has none	Open another view and come back to **Versions**
Download failed	The generated document has no file, or the download could not be prepared	Try again
Could not download this version. (or the service's reason) next to **Download** in **Versions**	The download could not be prepared	Try again; if it persists, contact support

Inbox

How to get here	**My Drive** › **Inbox**
Content status	Observed (2026-10-17, **Upload loose file** and **Link to a process**; **Archive** and **Discard** not walked). Field details checked against the Document Center interface source (`a8e9f24`) and service source (`d3ab083`) on 2026-10-22; not observed.

Header: **Inbox** — *Loose documents awaiting triage*.

Rows: origin (**Email**, **Upload**, **API**; another origin shows as stored), name, **Received** with date and time. Lists 50 at a time, newest first; **Load more (n of total)** adds the next 50. A file added with **Upload loose file** appears in the list as soon as the upload ends.

Buttons and actions

Button	What happens	Who can use it	Reversible?
Upload loose file	Opens the file chooser and adds one file to the Inbox (origin **Upload**); reads **Uploading…** meanwhile. Files can be up to 50 MB: a larger file is refused before it is sent.	Company administrator, technical profile	—
Load more	Adds the next 50 documents to the list	As above	—
Link	Opens Link to a process	As above	Not from the interface
Archive	Removes from the Inbox; status *Archived*; stays in its folder	As above	Not from the interface
Discard	Asks *Discard document? — This permanently removes the document.*; **Discard** deletes it, **Cancel** or **Esc** keeps it	As above	No

A row's buttons are disabled while one of its actions runs.

Common problems

Message	Cause	What to do
Loading inbox…	The list is loading	Wait
Nothing to triage — Loose documents (email, upload, API) will appear here	No document waits	Nothing to fix
Could not load the inbox with **Retry**	The list could not be loaded	Select **Retry**
Could not upload the file to the inbox. (or the service's reason), with **Dismiss**	**Upload loose file** failed; the list stays as it was	Add the file again; if it persists, contact support
…: the file is larger than 50 MB, the upload limit.	The chosen file is above the upload limit; nothing was sent	Use a smaller file
Could not archive … or *Could not discard …* (or the service's reason), with **Dismiss**	The action was refused or failed; the row stays in the list	Try again; if it persists, contact support

Link to a process dialog

How to get here	**Link** on a row of the *Inbox*
Content status	Observed on 2026-10-17 (suggestions and search); no document was linked. Field details checked against the Document Center interface source (`a8e9f24`) and service source (`d3ab083`) on 2026-10-22; not observed.

The dialog shows the document's name under **Link to a process**.

Section	Content
Suggested	Up to 5 processes whose identifier matches a CPF (11 digits) or CNPJ (14 digits), or whose e-mail matches an address, read from the document, with **by ID** or **by email**; each shows the requester (or process name), the form or workflow and *Received* with the date. *No automatic suggestion* when none.
Search process	*Name, requester or identifier*; up to 10 results, each with its name, form or workflow and requester; *No process found*
Link	Moves the document into the process; shows **Linking…**
Cancel / **Esc**	Closes without linking

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Name, requester or identifier	Searches processes by name, requester name or e-mail, or identifier (contains, ignoring case)	To search	Free text; an empty box does not search	Empty	—

Buttons and actions

Button	What happens	Who can use it	Reversible?
Search process (button) or **Enter**	Runs the search; *Searching…* meanwhile	Company administrator, technical profile	—

Button	What happens	Who can use it	Reversible?
Link on a suggestion or result	Moves the document into that process and removes it from the Inbox. All **Link** buttons are disabled while one runs.	As above	Not from the interface
Cancel, **Esc**, a click outside	Closes without linking	As above	—

Common problems

Message	Cause	What to do
Searching… under **Suggested**	Suggestions are loading	Wait
No automatic suggestion	Nothing read from the document matches a process	Use **Search process**
Could not load suggestions. Search for the process below.	The suggestions could not be loaded	Use **Search process**
No process found	Nothing matches the text	Try other words
Could not search processes.	The search failed	Try again
Could not link the document to the process. (or the service's reason) at the top of the dialog	The link was refused or failed; the dialog stays open	Try again; if it persists, contact support

Explorer

How to get here	**My Drive** › **Archived / Loose files**, or **Dashboard** › **Browse folders** or **Public shares**

Content status	Observed (2026-10-17, folder created, PDF uploaded, document opened). Field details checked against the Document Center interface source (`a8e9f24`) and service source (`d3ab083`) on 2026-10-22; not observed.

Element	What it does
Folder tree	**/ (root)** and the top-level folders, each with the number of folders inside it; ▸ opens a folder and ▾ closes it, at any depth. A folder without sub-folders shows ·.
Folder name, **Create**	Creates a folder under the selected folder (or the root); the tree reloads and shows it. Selecting **Create** again with the same name creates a second folder with that name.
Drop files here or click to choose	Uploads files into the selected folder; disabled with *Select a folder before uploading files.* until one is selected
Document cards	Name and status (*Draft*, *Review*, *Published*, *Archived*, *Purged*); open **Document details**

The selected folder is kept in the page address. The documents of the folder are listed 50 at a time; **Load more (n of total)** adds the next 50.

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Folder name	The new folder's name	Yes	Free text without `.` or `/` (checked by the service)	Empty	**Create** stays disabled while empty; when the service refuses the name its reason shows above the drop area

Field	What it does	Required	Format / values	Default	Validation messages
Files (drop area)	The files to upload	Yes	One or more files of any type, up to 50 MB each; the other service limits are those of the Dashboard upload	None	*Select a folder before uploading files.*; *…: the file is larger than 50 MB, the upload limit.*

Buttons and actions

Button	What happens	Who can use it	Reversible?
/ (root), a folder name	Selects the folder and lists its documents. At the root no documents are listed.	Company administrator, technical profile	—
▸ / ▾	Opens or closes a folder	As above	Yes
Create	Creates the folder	As above	Not from the interface
Drop area	Uploads each file as a new document with status *Draft*, one after the other; each appears in the list. A file that fails is reported and the next one is still uploaded; a document whose file did not arrive is removed again.	As above	No — the interface has no action to delete a document
Load more	Adds the next 50 documents of the folder	As above	—
A document card	Opens Document details	As above	—

Button	What happens	Who can use it	Reversible?
Try again	Loads the documents again	As above	—

Common problems

Message	Cause	What to do
Loading documents…	The documents are loading	Wait
No documents in this folder.	The folder is empty, or the root is selected	Select a folder
We couldn't load documents for this folder. with **Try again**	The documents could not be loaded	Select **Try again**
Select a folder before uploading files.	No folder is selected	Select a folder in the tree
Could not create the folder. or the service's reason above the drop area	The service refused the name (for example `.` or `/`) or the request failed	Correct the name and select **Create** again
A file name followed by the reason, or *Upload failed.*, above the drop area	That file was refused or failed (for example over the size limit or the monthly quota); the other files were still uploaded	Upload that file again once the cause is fixed
Could not load folders. with **Try again** in the tree	The folder list could not be loaded	Select **Try again**

Document details

How to get here	A document in **Explorer** or in search results

Content status	Observed (2026-10-17, preview, **Details**, versions, tags and **Timeline**; 2026-10-18, **Share**). Field details checked against the Document Center interface source (`a8e9f24`) and service source (`d3ab083`) on 2026-10-22; not observed.

Area	Content
Header	Name; *Status: … · Confidentiality: …* (*Public*, *Internal*, *Confidential*, *Highly confidential*); **Share**
Preview	Current version: PDF page by page, PNG and JPG. Others: *Preview not supported for this file type.*
Details	Status and confidentiality badges; **AI panel** (**Document type**, **Confidence** %, **Fraud**: *Possible fraud* / *No fraud detected*); **Versions** (`vN` , file name, size, ✓ *signed*); **Tags**
Timeline	Document events with their count, for example *Document created*, *Version uploaded*, *Viewed*, *Downloaded*, *Moved*, *Shared*, *Status changed*, *Signed*, *Archived*

Details, from the source.

- Status and confidentiality show in the interface language (in English: *Draft*, *In review*, *Published*, *Archived*, *Purged*; *Public*, *Internal*, *Confidential*, *Highly confidential*). Nothing on the screen changes them.
- The **AI panel** shows only when the document carries at least one of its values. **Document type** is the type the workflow recorded. **Confidence** is rounded to a whole percent: green from 80 %, amber from 50 %, red below. **Fraud** reflects only whether the tenant's fraud policy blocked the document (the `ai:fraudBlocked` tag): *Possible fraud* when it did, *No fraud detected* otherwise — including when no fraud check ran, so it is not the check's result. The graded result (*Trusted*, *Review recommended*, *Suspicious*) and its signals are in the workflow **Fraud review** queue and in Vision; see Fraud Detection.
- **Versions** lists every version with its size in KB. Downloading a version is possible only from the **Versions** view of Process details.

- **Tags** shows each tag as *key: value*, with × to remove it; the AI and system tags and the document type are left out. **—** when there is none. Under the tags, **Key** and **Value (optional)** with **Add tag** add a text tag; adding a key that already exists replaces its value. Keys starting with `ai:` or `sys:`, and `document_type`, belong to the automation and cannot be added here.
- The **Timeline** lists up to the 50 most recent events, each with its name, date and time and the user's identifier. Other events it can show: *Renamed*, *Tag added*, *Tag removed*, *Permissions changed*, *Share link revoked*, *Checked out*, *Checked in*, *Purged*, and the review events of a process (*Documents refused*, *Returned to requester*, *Notification attempt*, *Requester resent documents*, *Reprocessing started*, *Requester reminded*, *Correction escalated*, *Correction link expired*, *Document decided*, *Review closed*).

Fields

Create share link dialog: *The recipient email is only recorded for your records and audit. It does not restrict who can open this link; to limit access, set a password, an expiry date or a maximum number of accesses below.*

Field	What it does	Required	Format / values	Default	Validation messages
Recipient email for records (optional)	Kept for your records and audit; it does not restrict who can open the link	No	An e-mail address, for example *recipient@example.com*	Empty	The browser's e-mail check
Password (optional)	The person opening the link	No	Hidden text, sent exactly as typed	Empty	—

Field	What it does	Required	Format / values	Default	Validation messages
	must type it				
Expires on (optional)	The link stops working at this date and time	No	Date and time, in the future	Empty — no expiry	*Enter a valid expiry date and time.*; *The expiry must be in the future.*
Maximum number of accesses (optional)	How many times the link can be opened	No	Whole number from 1	Empty — no limit	*Enter a whole number of 1 or more.*

Every link created here allows downloads.

Buttons and actions

Button	What happens	Who can use it	Reversible?
Details, **Timeline**	Switch the side panel	Company administrator, technical profile	—
‹, **›**, *Page … of …*	Turn the pages of a PDF in the preview	As above	—
Share	Opens **Create share link**	As above	—

Button	What happens	Who can use it	Reversible?
Create link	Checks the fields, creates the link, copies it and shows **Link copied to clipboard** with the link, **Copy again** and **Dismiss** for 5 seconds. The *Shared* event is added to the document and the link appears under **Active share links**. Disabled while the link is being created.	As above	Yes — **Revoke**
Revoke (under **Active share links**)	Asks *Revoke this link? — Everyone with this link loses access at once, including people who have already opened it. This cannot be undone.* Confirming revokes the link: its address never opens again, the sessions already opened from it stop working, and the *Share link revoked* event is added to the document. The row leaves the list and *Link revoked.* is shown. A failure shows the service's reason and keeps the row.	As above	No
Cancel, a click outside	Closes the dialog and clears its fields	As above	—
Copy again	Copies the link again	As above	—
Dismiss	Hides the link	As above	—
Add tag	Adds the tag (or replaces the value of an existing key); the *Tag added* event is recorded	As above	Yes — **×** removes it

Button	What happens	Who can use it	Reversible?
× on a tag	Removes the tag; the *Tag removed* event is recorded	As above	Yes — add it again
Try again	Loads the document again after a failed load	As above	—

Active share links, under the header, lists the document's links that still open: the recipient email for records (*No recipient recorded* when none), the expiry (*Expires …* or *No expiry*), how many times it was opened (*Opened … of … times* when it has a limit) and *Password* when it has one. Revoked, expired and used-up links are left out; *No active share links.* when there is none, *Could not load the share links.* when the list fails. The link address itself is shown only once, when it is created.

How the recipient opens the link — as many times as it is valid for, each opening a session of one hour — is described under *Shared document* in the user manual's *Document Center screens*.

Common problems

Message	Cause	What to do
Could not load this document. (or the service's reason) with **Try again**	The document does not exist or could not be loaded	Select **Try again**; if it persists, go back and open it again, or contact support
Preview unavailable	The preview could not be prepared	Reload the page
Preview not supported for this file type. with *Supported: PDF, PNG, JPG. Other formats (DOC, XLS, etc.) on the future roadmap.* and **Download**	The file is not a PDF, PNG or JPG	Use **Download**

Message	Cause	What to do
Enter a valid expiry date and time., *The expiry must be in the future.*, *Enter a whole number of 1 or more.*	A share field is invalid	Correct the field
Could not create the share link. (or the service's reason) in the dialog	The service refused or failed to create the link; the dialog stays open with the fields kept	Correct the fields or try again; if it persists, contact support
Could not save the tag., *Could not remove the tag.* (or the service's reason)	The tag change was refused or failed	Try again
Could not load the timeline.	The events could not be loaded	Reload the page
No events yet.	The document has no recorded event	Nothing to fix

Earlier note, resolved: on 2026-10-18 the dialog text suggested a password, an expiry date or a maximum number of accesses but offered none of them (XTRAK-2592). The fields were added in Document Center interface `f6057c5`; not yet observed.

Search documents

How to get here	The search box at the top (*Search processes and documents*), or **Dashboard** › **Search documents**
Content status	Observed (2026-10-18, a word in the names of synthetic documents). Field details checked against the Document Center interface source (`a8e9f24`) and service source (`d3ab083`) on 2026-10-22; not observed.

Element	Content
Search box	*Try: RG sent today, contracts this month…*; **Search**
Understood as:	How the phrase was read: the text in quotes and the period (**Today**, **Yesterday**, **This week**, **This month**, **This year**, **Last n days**)
Filters	Values to narrow the results, each with its count, in four groups: **File type** (for example *PDF*), **Status**, **Confidentiality** and **Tags** (documents that carry a tag with that key, whatever its value).
Active filters:	One chip per selected value, *Group: value ×*; the chip removes it. **Clear filters** removes them all.
Result	Name, file type, highlighted passage, date, *score*

Before searching: *Search by name, content or a phrase like "invoices this week".* No match: *No results.*

Fields

Field	What it does	Required	Format / values	Default	Validation messages
Search box	Searches the documents' names, content and tag values. One period phrase is recognised and turned into a date range on when the file was uploaded; the other words are searched as text.	No — an empty search is sent without a text filter	Free text. Period phrases, in English or Portuguese: *today* / *hoje*, *yesterday* / *ontem*, *this week* / *esta semana*, *this month* / *este mês*, *this year* / *este ano*, *last … days* / *últimos … dias* (up to 3 digits). A week starts on Monday.	Empty, or the text typed in the top search box	—
A value under **Filters**	Narrows the results to that value; selecting it again removes it. One value per group; values in different groups all apply. A selected value stays listed, so it can always be removed.	No	The values the service returns with the results	None	—

Buttons and actions

Button	What happens	Who can use it	Reversible?
Search or **Enter**	Runs the search from the first page; disabled while it runs. Results show 20 per page with their total (*n results*).	Company administrator, technical profile; results respect each folder's permissions	—
A value under **Filters**, a chip under **Active filters:**, **Clear filters**	Adds or removes filters and runs the search again from the first page	As above	Yes
Previous / **Next**, *Page … of …*	Turn the result pages	As above	—
Try again	Runs the failed search again	As above	—
A result	Opens Document details	As above	—

Common problems

Message	Cause	What to do
No results.	Nothing matches the text, the period or the selected filters	Use fewer words, a wider period, or remove the filters
The search failed. Try again. (or the service's reason) with **Try again**	The search failed	Select **Try again**; if it persists, contact support

Not observed (all screens): what another profile sees when it opens the addresses behind *My Drive* other than *My Drive* itself; the display of a **Failed** process; **Archive** and **Discard** in the *Inbox*.

Last reviewed 2026-10-23 · Version 1.2.0 · Owner: product-documentation

Chapter versions

Chapter	Version	Last reviewed
Start here	`1.0.0`	2026-10-16
Manage members and profiles	`0.1.0`	2026-10-08
Portal journey languages	`1.0.0`	2026-10-18
Workflow administration overview	`1.0.0`	2026-10-09
Design and publish a workflow	`1.1.1`	2026-10-09
Organize human review	`1.1.0`	2026-10-22
Validate digital signatures	`1.0.0`	2026-10-09
Connect your systems	`1.1.0`	2026-10-09
Workflow phases	`1.0.0`	2026-10-22
Start	`1.0.0`	2026-10-22
Crop	`1.0.0`	2026-10-22
Split	`1.0.0`	2026-10-22
Classification	`1.0.0`	2026-10-22
Extraction	`1.0.0`	2026-10-22
Zero Shot	`1.0.0`	2026-10-22
Facematch	`1.0.0`	2026-10-22
Cross-check	`1.1.0`	2026-10-22
Manual Review	`1.0.0`	2026-10-22
Conditional	`1.0.0`	2026-10-22

Chapter	Version	Last reviewed
AI Executor	`1.0.0`	2026-10-22
Send Message	`1.0.0`	2026-10-22
Generate document	`1.0.0`	2026-10-22
Signature	`1.0.0`	2026-10-22
Signature validation	`1.0.0`	2026-10-22
REST integration	`1.0.0`	2026-10-22
Send	`1.0.0`	2026-10-22
KYB	`1.0.0`	2026-10-22
REST integration — overview	`1.0.0`	2026-10-22
REST integration — Request	`1.0.0`	2026-10-22
REST integration — Schemas and mapping	`1.0.0`	2026-10-22
REST integration — Result, Test and Publication	`1.0.0`	2026-10-22
REST integration — messages	`1.0.0`	2026-10-22
Workflow administration screens	`1.1.0`	2026-10-09
Vision administration overview	`1.0.0`	2026-10-22
Dashboard	`1.0.0`	2026-10-22
Templates	`1.0.0`	2026-10-22
Template editor — Schema	`1.0.0`	2026-10-22
Redaction policy	`1.0.0`	2026-10-22
Template agent	`1.0.0`	2026-10-22

Chapter	Version	Last reviewed
Playground	`1.0.0`	2026-10-22
Tenant agent settings	`1.0.0`	2026-10-22
Developers	`1.0.0`	2026-10-23
MCP	`1.0.0`	2026-10-22
Template administration overview	`1.0.0`	2026-10-22
Templates	`1.0.0`	2026-10-22
Field sources	`1.0.0`	2026-10-22
Render playground	`1.0.0`	2026-10-22
Word add-in	`1.0.0`	2026-10-22
Document Center administration overview	`1.0.0`	2026-10-18
Triage the Inbox	`1.0.0`	2026-10-17
Follow a process	`1.0.0`	2026-10-18
Organize and share documents	`1.0.1`	2026-10-22
Search documents	`1.0.0`	2026-10-18
Document Center administration screens	`1.2.0`	2026-10-23