ABBYY Vantage Connector for Camunda 8

Automate document capture and data extraction in Camunda using the ABBYY Vantage Connector.

By: Deepak Goyal (Partner Innovation and Engagement Consultant At ABBYY)

Table of Contents

Introduction

The ABBYY Vantage Connector for Camunda brings advanced Document AI directly into Camunda’s process orchestration platform. ABBYY Vantage transforms unstructured documents into accurate, structured data, while Camunda orchestrates decisions and routes work across systems and teams. Together, they deliver scalable, end-to-end automation for document-centric processes such as claims handling, onboarding, financial operations, compliance, and customer service.

The Connector provides seven prebuilt operations that handle the complete document-processing lifecycle, including creating transactions, uploading files, start processing, checking status and retrieving results. Each operation is packaged as a reusable subprocess so workflow designers can configure logic once and reuse it in any number of workflows without additional integration work. This modular approach delivers clean orchestration, consistent error handling and predictable outputs.

The prebuilt connector offers a repeatable, secure and no-code integration that accelerates time to value and modernizes enterprise workflows.

Key Features

✓ No-code integration
ABBYY Vantage API calls are available through a reusable Connector Template.

✓ Full coverage of core Vantage Transaction Operations
This includes document upload, transaction lifecycle, processing start, polling, and downloads.

✓ Reusable across all workflows
Configure once and re-use in any BPMN process.

✓ Secure authentication
Supports OAuth 2.0 Client Credentials flow using Camunda Secrets.

✓ Works in Camunda SaaS
Scales automatically for enterprise workloads.

Possible Use Cases

The ABBYY Vantage Connector for Camunda can be used in a wide range of document-driven automation scenarios across industries. Below are some common use cases.

Financial Services Documents

Automate the processing of high-volume financial documents directly inside Camunda workflows. Examples:

- Bank statements
- Tax forms
- Lending and credit documents

Know Your Customer (KYC)

Automate identity and verification document processing for customer onboarding and compliance checks. Examples:

- ID proofs
- Address proofs
- Customer application forms

Loan & Lending Processing

Use the connector to process multiple types of loan-related documents in a single end-to-end workflow. Examples:

- Loan applications
- Income documents
- Property and collateral documents

Insurance Claims Processing

Automate the intake and processing of insurance claim documents and supporting records. Examples:

- Insurance claims Form
- Claim Supporting documents

Transportation & Logistics

Digitize and automate shipping and logistics documentation within orchestrated workflows. Examples:

- Shipping instructions
- Waybills
- Receipts or Orders

Finance & Accounting

Automate invoice and finance document processing and route extracted data to backend systems. Examples:

- Invoices or Utility bills
- Expense documents
- Vendor onboarding

Prerequisites

You need the following before using the connector:

ABBYY Vantage

- An active Vantage environment & License
- OAuth Client ID & Client Secret
- Base URL (e.g., https://vantage-eu.abbyy.com/)

Camunda 8

- Camunda SaaS account
- Project and cluster
- At least **three secrets** created:
  ```
  VANTAGE_BASE_URL
  ABBYY_CLIENT_ID
  ABBYY_CLIENT_SECRET
  ```

Authentication

The ABBYY Vantage Connector uses OAuth 2.0 Client Credentials to securely authenticate with ABBYY Vantage APIs. All authentication values must be configured using Camunda Secrets.

Before using the connector, ensure that the required credentials are created in Camunda and obtained from your ABBYY Vantage environment.

Required Authentication Fields

Field	Description	Where to Find It
baseUrl	The base URL of your ABBYY Vantage API environment	Provided by ABBYY during environment setup.
oauthTokenEndpoint	The OAuth endpoint used to request the access token	Available in ABBYY Vantage developer documentation also it is preconfigured in Vantage connector
clientId	The OAuth Client ID used to authenticate API requests	Generated in ABBYY Vantage under Configuration-Public API Client
clientSecret	The OAuth Client Secret associated with the Client ID	Generated in ABBYY Vantage under Configuration-Public API Client
clientAuthentication	Defines how credentials are sent to the token endpoint.	ABBYY Vantage documentation. Always "`basicAuthHeader`"
scopes	The permission scope used for accessing ABBYY Vantage APIs	ABBYY Vantage documentation. preconfigured in connector

Example Configuration

`baseUrl`: https://vantage-eu.abbyy.com/
`oauthTokenEndpoint`: https://vantage-eu.abbyy.com/auth2/connect/token
`clientId`: {{secrets.ABBYY_CLIENT_ID}}
`clientSecret`: {{secrets.ABBYY_CLIENT_SECRET}}
`scopes`: openid permissions global.wildcard

Connector Operations

This connector supports **seven** ABBYY Vantage operations.

Operation 1: Create Transaction

(Used when you want a multi-step flow: create → add files → start)

How is this operation used?
Through a BPMN **support subprocess** (e.g., ABBYY Vantage: Launch Transaction in sample flow).
The subprocess calls the connector operation **"Creates a processing transaction"**.

Runtime Inputs (from BPMN subprocess)
Only one meaningful business input:

- **skillId** – The Vantage Skill ID selected by the process designer.

No need to configure `baseUrl, clientId, clientSecret,` they come from secrets via connector.

Request Body Created by the Subprocess
The subprocess builds a JSON body containing:

- `skillId`
- Optionally:
  `generateMobileInputLink`
  `registrationParameters[]`
  `skillParameters[]`

Typical Output Returned to BPMN

```
transactionId
mobileInputLink (only when enabled)
```

What the operation does
Creates an empty transaction container in Vantage.

Operation 2: Add Files to Transaction

How is this operation used?
Through a BPMN **support subprocess** (e.g., ABBYY Vantage: Launch Transaction in sample flow).
This subprocess calls connector operation: "Adds files to the transaction"

Runtime Inputs

- `transactionId` – From previous step
- **raw files** – As Camunda documents
  (`storeId` + `documentId` provided directly by Camunda Document Store)

Multipart Body Created by Subprocess

- `Files[]` (binary files)
- Optional metadata (auto-filled if not configured)
  `registrationParameters`
  `imageProcessingOptions`
  `index`

Output
Vantage upload acknowledgement

What the operation does
Uploads documents into an already created transaction.

Operation 3: Start Transaction

How is this operation used?
Through a BPMN support subprocess (e.g., ABBYY Vantage: Launch Transaction in sample flow).
Calls connector operation: "Starts the transaction"

Runtime Input

- `transactionId`

Output

- Simple "transaction started" acknowledgment

What the operation does
Moves the transaction from New → Ready so that processing can begin.

Operation 4: Get Transaction Information

How is this operation used?
Through a BPMN support subprocess (e.g., ABBYY Vantage: Check Transaction Status in sample flow).
This subprocess repeatedly calls the connector operation: "Gets transaction info"

Runtime Inputs

- `transactionId`

Output Returned to BPMN

- `transactionInfo` containing:
  - Status
  - Documents
  - Extracted result files
  - Manual Review Link

- Errors if any
- Source Files

What the operation does
Retrieves where the transaction is in the workflow cycle (Processing, Processed, Failed, etc.)

Operation 5: Download the Result Files

How is this operation used?
Through a BPMN support subprocess (e.g., ABBYY Vantage: Download Results in sample flow).
This subprocess calls connector operation: “Downloads the result files”

Runtime Inputs

- `transactionInfo` (passed directly into subprocess)
- The subprocess extracts:
  `transactionId`
  `fileIds` for result files

Output
Mapped into:

- `resultFilesVantage`
  (using template expression `flatten(outputFiles)`)

Each result file is stored as a Camunda document:

- JSON
- PDF
- TIFF
- PNG
- Fields.json
  (depending on Skill output)

What the operation does
Retrieves extraction results after Vantage completes processing.

Operation 6: Start Processing of the Files in the Transaction (Launch Transaction)

(Used when you want a single file processing with a single file per transaction input)

How is this operation used?
This subprocess calls connector operation: “Starts processing of the files in new transaction” (API: /transactions/launch)
Runtime Inputs
From your template:

- `vantageSkillId` (Skill ID)
- `rawFilesVantage` (optional if you include files in launch)

Request Body / Multipart Created by Subprocess

- `skillId`
- `Files[]`
- Optional settings
  `registrationParameters[]`
  `skillParameters[]`

Output

- `transactionId` (always returned)

What the operation does?
Creates a transaction, uploads files, and starts processing in one call.

Operation 7: Download the Source File of the Transaction

How is this operation used?
Read `transactionInfo` then calls the connector operation "Downloads the source file of the transaction".

Runtime Inputs
`transactionId` — transactionInfo.id (required).
`documentId` — a document.id selected from `transactionInfo.documents[]`
`fileId` — the source file id taken from `transactionInfo.sourceFiles[]`

Output
A Camunda document variable containing the downloaded binary and metadata. Typical mapped fields (use `storeResponse` = true in the connector so response.document.* is available):

```
documentId (Camunda document id)
storeId
contentHash
metadata.fileName
metadata.size
```

What the operation does
Retrieves the raw file exactly as uploaded to Vantage.

BPMN Usage Example

This example follows the main process in VantageIntegrationDemo.bpmn and uses the three subprocesses you provided. It shows variables and bindings exactly as used in the demo.

Upload file (form)

- User uploads a file using the form in the main process.
- The uploaded file is stored in the process variable `rawFilesVantage`.

Call Activity - launchTransaction (subprocess)

- Inputs (main → subprocess):
  - `rawFilesVantage` (the uploaded file)
  - `vantageSkillId` (from secret)
- What happens inside Launch Transaction:
  - Create a transaction in Vantage.
  - Add the `rawFilesVantage` to the transaction.
  - Start the transaction.
- Output (subprocess → main):
  - `transactionId` (returned from Vantage)

Call Activity - checkTransactionStatus (subprocess)

- Inputs:
  - `transactionId` = the `transactionId` from `launchTransaction`
  - `waitDuration` (poll interval)
  - `maxPolls` (maximum attempts)
- What happens inside Check Transaction Status:
  - Poll Vantage using `transactionId` until processing finishes or fails.
  - On completion (Processed / Failed / Requires Manual Review) the subprocess returns the full transaction JSON.
- Output:
  - `transactionInfo` — full transaction response stored in the main process

Call Activity - downloadResults (subprocess)

- Input:
  - `transactionInfo` = the `transactionInfo` returned by checkTransactionStatus
- What happens inside download Results:
  - Read `transactionInfo.documents[]` and `resultFiles` entries.
  - Download each result file and store them as Camunda documents.
- Output:
  - `resultsFileVantage` — a flattened list of downloaded result-file objects (used downstream)

Post-processing (example use)

- Use the resultsFileVantage variable as needed. For example:
    - Attach resultsFileVantage to an email and send to recipients.
    - Store files in an archive.
    - Feed extracted data into downstream systems.

Notes and reminders

- `skillId` is taken from secrets, not typed at runtime.
- `documentId` and `fileId` (if you need original source files) are available inside `transactionInfo` (look in `transactionInfo.documents[]` and `transactionInfo.sourceFiles[]`).
- Ensure `checkTransactionStatus` stores the entire response to `transactionInfo` so `downloadResults` can find result file ids.

Best Practices

Use dedicated subprocesses
Keep the workflow modular by letting each operation run inside its own subprocess such as launch, status check, and results download. This keeps the solution clean, reusable, and easy to maintain.

Enable storeResponse = true whenever possible
For download operations this is required so the binary file can be saved as a Camunda document.
For other operations such as Create Transaction, Add Files, and Start Transaction, enabling storeResponse is also recommended because it helps with debugging and troubleshooting.

Use secrets for all configuration values
Store the Vantage base URL, client ID, client secret, and any environment-specific settings in Camunda Secrets. Never hardcode sensitive information inside the BPMN model.

Troubleshooting

Error	Cause	Fix
401 – Unauthorized	Invalid OAuth credentials, missing client ID/secret, or incorrect token endpoint.	Check Camunda secrets (ABBYY_VANTAGE_CLIENT_ID, ABBYY_VANTAGE_CLIENT_SECRET, ABBYY_VANTAGE_BASE_URL). Ensure the OAuth token endpoint matches your environment.
404 – Transaction not found	The transactionId, documentId or fileId passed to the API does not exist in Vantage	Verify that IDs are extracted correctly from transactionInfo. Confirm the transaction exists and is not deleted.
400 – Invalid parameters or missing required fields	Happens when request body is malformed, files exceed limits, or required fields (like skillId) are missing.	Check the subprocess mappings, verify file sizes, and confirm required fields (e.g., skillId) are passed correctly.
Empty download file	The connector did not store the binary HTTP response because storeResponse was disabled.	Enable storeResponse = true in all download operations (result and source files).
Processing stuck / never completes	Vantage is still processing, or no polling loop exists. The API will keep returning status = Processing until completed.	Ensure the ABBYY_Vantage_Check_Transaction_Status subprocess is used. Increase maxPolls or waitDuration if documents take longer to process.
409 – Max file count exceeded	Attempted to upload more than the allowed number of files.	Reduce the number of files uploaded in one request or split uploads across multiple transactions.

Limitations

- The connector does not provide operations for creating, training, or publishing Vantage Skills. All Skills must already exist in the Vantage environment.
- The integration relies on the ABBYY Vantage Public API v1 as defined in the Swagger specification.
- Camunda Secrets must be configured for authentication. The connector cannot operate without valid OAuth credentials stored as secrets.
- Multipart file uploads must follow the structure used in the provided subprocess templates to ensure Vantage accepts the files correctly.
- The solution is supported only in Camunda 8 SaaS or self-managed environments that provide file-variable support for handling document uploads and downloads.
- Vantage transaction must be initiated in Camunda. If the document is initiated from any other source, we cannot process the result using this connector.