



Document Loop and Queue

Punchout Documentation

Guide for Magento administrators, technical project owners, and implementation teams.

LAST UPDATED
2026-09-26

SOURCE
[Open this article online](#)

EXTENSION
[Magento 2 Extension Punchout](#)

| Document Loop and Queue

The document loop starts with an approved cXML **OrderRequest**. Magento can record, validate or create an order from that request. When fulfilment begins, Magento can send a **ConfirmationRequest**, **ShipNoticeRequest** and **InvoiceDetailRequest** to the procurement system.

Order creation and returned-cart pricing are covered in **Inbound Orders and Pricing**. This page focuses on outbound documents and their queue.

Configure outbound cXML identity

Generated cXML uses the effective supplier credential domain, identity and shared secret retained for the inbound order. Configure those values before enabling **ConfirmationRequest** delivery or testing shipment and invoice callbacks. Buyer-profile settings override Store-view defaults.

Configure a separate endpoint for each document type: Confirmation URL, ASN URL and Invoice URL. The browser cart-return URL is used only to return the shopping cart and is never reused for later server-to-server documents.

Magento generates cXML 1.2.070 envelopes for confirmation, shipment and invoice documents. The destination must be a validated HTTPS URL.

ConfirmationRequest

When **Send cXML ConfirmationRequest** is enabled, the effective Confirmation Mode decides how the document is prepared.

- **Automatic Whole-Order Confirmation** confirms a successfully created order without waiting for an administrator. A cancellation produces the corresponding rejection.
- **Manual Line Confirmation** waits for an administrator to decide every order line in **Punchout > Inbound Orders**.

For a manual confirmation, open the inbound order and select **Accept**, **Reject** or **Backordered** for every line. Accepted and backordered lines may include an expected delivery date. Rejected lines cannot have a delivery date. Every inbound line must have exactly one decision before the confirmation can be submitted.

Submitting the form creates one immutable confirmation document in the queue. The same inbound order cannot create a second confirmation. Its expected-delivery values are shown in the Admin user's locale, while the cXML document uses the protocol's ISO date format.

Inbound Order PO-MCP-LINE-20260823-003 ← Back

Replacement Magento Order ID

Order Update Progress

Attempts: 1

Maximum Attempts: 5

Failure Code

Failure Message

Order Lines

20 per page 1 of 1

External Line	Supplier Part ID	Description	Quantity	Unit of Measure	Confirmation	Expected Delivery
1	24-MB01	Joust Duffle Bag	1	PCE	Accept	Leave empty for unchanged or unknown delivery date. Rejected lines cannot have a date.

Technical Payloads

Copyright © 2026 Magento Commerce Inc. All rights reserved.

Magento ver. 2.4.8-p5
Hyvä Theme Module ver. 1.4.3
[Privacy Policy](#) | [Account Activity](#) | [Report an Issue](#)

A delivery failure remains visible with the inbound order and queue entry so an administrator can diagnose the receiver before retrying.

Automatic outbound documents

When Magento saves a shipment or invoice, the module checks whether its order quote is linked to a PunchOut context.

- A shipment creates one **asn_outbound** queue row containing the shipment number, tracking information and shipped SKU quantities.
- An invoice creates one **invoice_outbound** queue row containing the invoice number, order references, item details, summary amounts and order currency.
- The same Magento entity is not queued twice for the same document type.

The queue builds the outbound document once, encrypts the exact payload at rest and sends it to the document-specific HTTPS endpoint. Retries reuse the same retained payload and routing data. Shipment and invoice delivery therefore remains independent of the short-lived browser context.

Queue processing

DASHBOARD

SALES

CATALOG

CUSTOMERS

MARKETING

CONTENT

REPORTS

STORES

SYSTEM

FIND PARTNERS & EXTENSIONS

This is only a demo store. You can browse and place orders, but nothing will be processed.

Punchout Document Queue

Demo expiry: 2026-08-20 08:57 UTC (4d 22h left)

Filters

Default View

Columns

Queue ID

Created At

Entity ID

Attempts

Max Attempts

Next Attempt At

Document Type

Entity Type

Status

Last Error

Cancel

Apply Filters

Actions

2 records found

20 per page

1 of 1

Queue ID	Created At	Document Type	Entity Type	Entity ID	Status	Attempts	Max Attempts	Next Attempt At	Last Error	Actions
2	2026-08-15 12:06:04	asn_outbound	shipment	4	pending	0	5	2026-08-15 12:06:04		Select
1	2026-08-15 12:05:44	invoice_outbound	invoice	4	pending	0	5	2026-08-15 12:05:44		Select

Copyright © 2026 Magento Commerce Inc. All rights reserved.

Magento ver. 2.4.8-p5

Privacy Policy | Report an Issue

Queue item #5 has been executed.

Filters

Default View

Columns

Actions

5 records found

20 per page

1 of 1

Queue ID	Created At	Document Type	Entity Type	Entity ID	Status	Attempts	Max Attempts	Next Attempt At	Last Error	Archive State	Archived At	Actions
5	2026-08-22 14:38:08	Outbound Ship Notice	shipment	4	Completed	1	5			Active		View
4	2026-08-22 14:37:42	Outbound Invoice	invoice	4	Completed	1	5			Active		View
3	2026-08-22 14:29:06	Outbound Confirmation	inbound_order	3	Completed	1	5			Active		View
2	2026-08-22 13:19:15	Outbound Ship Notice	shipment	3	Delivery Unknown	1	5		Document delivery outcome is unknown for HTTP status 0.	Active		Select
1	2026-08-22 13:19:15	Outbound Invoice	invoice	3	Delivery Unknown	1	5		The receiver returned an invalid cXML acknowledgement.	Active		Select

Each queue detail keeps the business entity, buyer, attempt count, safe request summary and delivery result together. The examples below show the three outbound document types from one completed order lifecycle; each receiver returned HTTP 204.

Back

Reset

Force Run

Queue ID

3

Status

completed

Document Type

confirmation_outbound

Entity

inbound_order #3

Correlation ID

confirmation-dda616273428fcf170970938

Buyer ID

CEDARSTONE-CXML-US

Attempts

1 / 5

Next Attempt At

Last Error

Created At

2026-08-22 12:29:06

Updated At

2026-08-22 12:37:16

Request Payload

{ "order_id": 4, "confirmation_type": "accept", "confirmation_payload_id": "confirmation-dda616273428fcf170970938", "confirmation_timestamp": "2026-08-22T12:29:06Z", "external_order_id": "CEDARSTONE-PO-2026-0808", "source_

Response Payload

{ "success": true, "delivery_state": "delivered", "http_status": 204, "cxml_status_code": null, "cxml_status_text": "", "response_excerpt": "", "message": "Document delivered successfully." }

		Back Reset Force Run
Queue ID	4	
Status	completed	
Document Type	invoice_outbound	
Entity	Invoice #4	
Correlation ID	cedarstone-po-2026-008@cedarstone.example	
Buyer ID	CEDARSTONE-CXML-US	
Attempts	1 / 5	
Next Attempt At		
Last Error		
Created At	2026-08-22 12:37:42	
Updated At	2026-08-22 12:43:18	
Request Payload		
{ "invoice_number": "000000004", "invoice_total": 103, "currency_code": "EUR", "store_id": 1 }		
Response Payload		
{ "success": true, "document_type": "invoice_outbound", "document_number": "000000004", "invoice_total": 103, "currency": "EUR", "http_status": 204, "response_excerpt": "", "cxml_status_code": null, "cxml_status_text": "", "message": "" }		

		Back Reset Force Run
Queue ID	5	
Status	completed	
Document Type	asn_outbound	
Entity	shipment #4	
Correlation ID	cedarstone-po-2026-008@cedarstone.example	
Buyer ID	CEDARSTONE-CXML-US	
Attempts	1 / 5	
Next Attempt At		
Last Error		
Created At	2026-08-22 12:38:08	
Updated At	2026-08-22 12:44:07	
Request Payload		
{ "shipment_id": "000000004", "tracking_number": "", "items": [{ "sku": "24-U002", "quantity": 1 }, { "sku": "24-U005", "quantity": 1 }, "store_id": 1 }		
Response Payload		
{ "success": true, "document_type": "asn_outbound", "document_number": "000000004", "line_count": 2, "tracking_number": "", "http_status": 204, "response_excerpt": "", "cxml_status_code": null, "cxml_status_text": "", "message": "" }		

The `punchout_document_queue_process` job runs in Magento's `default` cron group every five minutes. It processes due `pending` rows per store, using the store-scoped batch size and retry settings.

For queue inspection and controlled document-type processing, see [Operations and CLI](#).

Statuses are:

- `pending`: ready now or waiting until `next_attempt_at`;
- `processing`: claimed by a worker or Force Run;
- `completed`: the document endpoint answered successfully;
- `failed`: the maximum attempts were used.
- `delivery_unknown`: Magento cannot determine whether the remote system received the document, so automatic retry is stopped to avoid a duplicate business document.

Network exceptions and non-2xx responses count as failures. A row below its maximum returns to `pending` with the configured delay.

An interrupted delivery can produce `delivery_unknown` when the remote outcome is ambiguous. Review the receiving system first. Only an administrator with the dedicated permission can deliberately schedule that row again.

✖ Queue item could not be executed.

Queue ID	Created At	Document Type	Entity Type	Entity ID	Status	Attempts	Max Attempts	Next Attempt At	Last Error	Archive State	Archived At	Actions
2	2026-08-22 13:19:15	Outbound Ship Notice	shipment	3	Delivery Unknown	1	5		Document delivery outcome is unknown for HTTP status 0.	Active		Select
1	2026-08-22 13:19:15	Outbound Invoice	invoice	3	Pending	0	5	2026-08-22 13:19:15		Active		Select

Confirmation delivery is durable. When the effective configuration requires a confirmation, the inbound-order record retains the immutable supplier routing and the queue can recover a missing confirmation intent after a worker or deployment interruption. The synchronous **OrderRequest** response does not wait for a remote confirmation endpoint. This keeps order acknowledgement separate from later server-to-server document delivery while still leaving the intent visible in **Document Queue**.

Work with a failed row

Open **Punchout > Document Queue**, then View the row.

Queue ID	Created At	Document Type	Entity Type	Entity ID	Status	Attempts	Max Attempts	Next Attempt At	Last Error	Archive State	Archived At	Actions
2	2026-08-22 13:19:15	Outbound Ship Notice	shipment	3	Failed	1	5		PunchOut ASN endpoint is not configured.	Active		Select
1	2026-08-22 13:19:15	Outbound Invoice	invoice	3	Pending	0	5	2026-08-22 13:19:15		Active		Select

- **Force Run** schedules the row immediately and processes it once. It does not erase the existing attempt count.
- **Reset** clears attempts, response and error data and returns the row to **pending**. Use it only after the underlying configuration or receiver problem has been corrected.
- **Retry Unknown Delivery** deliberately reschedules a **delivery_unknown** row after the receiving system has been checked.
- **Delete** removes the operational record and should not be used as a substitute for diagnosing delivery.

The module and queue must be enabled for the row's Store view, the retained supplier identity must be complete, and the document-specific URL must accept HTTPS POST requests. Read the saved HTTP response or transport error before resetting the attempt count.

Direct document endpoints

/punchout/document_asn/outbound and **/punchout/document_invoice/outbound** execute the same document delivery handlers immediately and return their result. They use the effective document-specific endpoint and do not create a Magento shipment or invoice.

Send a client-generated **X-Punchout-Request-Key** for OrderRequest, ASN and invoice calls. Reuse it only for the same semantic request. Magento replays the saved result for an exact retry and rejects changed input that reuses the key.