



Queue and cron operations

E-Invoice 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 E-Invoice](#)

Queue and cron operations

Choose one processing mode per document flow and monitor the worker that owns it. A **queued** or **scheduled** row is a work item, not a successful document.

Queue mode

Invoice and credit memo messages use separate DB-backed topics and consumers:

```
php bin/magento queue:consumers:start mageb2b.einvoice.generate
php bin/magento queue:consumers:start
mageb2b.einvoice.generate.creditmemo
```

Run both under the same process supervision used for other Magento consumers. The invoice consumer does not process credit memos. A successful consumer changes **queued** to **generated**, meaning that local document generation completed; it does not prove delivery. An exception stores **error** and the message can enter the queue backend's failure path.

Queue verification workflow

1. Configure **Message Queue** for only the document flow under test.
2. Start the matching supervised consumer and confirm Magento lists it.
3. Generate a new invoice or credit memo and note its numeric entity ID.
4. Confirm that the document first enters **queued** and that the matching topic receives work.
5. Wait for the consumer, then reopen the detail page. Expect **generated** with local file metadata or **error** with a stored message.
6. Download and validate the result. Verify delivery separately through the intended downstream system.

Scheduled mode and the cron worker

A single job in Magento's **default** group handles both document flows:

Job	Schedule
einvoice_process_scheduled	minute 00 every hour

For each store, the worker processes up to the flow's **Scheduled Batch Size** (default **100**, allowed range 1-1000) of due **scheduled** rows. Invoice and credit-memo flows have separate batch settings.

The same worker also performs stale-claim recovery: **queued** and **processing** rows older than the flow's **Recovery Timeout (Minutes)** (default **30**, allowed range 5-1440)

are claimed again and reprocessed. This rescues documents stranded by a crashed worker or consumer. It does not retry rows already marked **error** — those require a deliberate manual regeneration.

Diagnosis sequence

1. Confirm the document's store view and exact generation mode.
2. For queue mode, list consumers and inspect the queue backlog.
3. For scheduled mode or stuck rows, inspect **cron_schedule** for **einvoice_process_scheduled** and its latest status.
4. Run `php bin/magento cron:run --group=default` once as a diagnostic, not as the production scheduler.
5. Read the stored error before changing source data or retrying.
6. Fix deterministic data or configuration errors, then retry once through the detail-page action or REST **force**.

Capture queue evidence from the consumer/process view, the affected document detail page and the queue backend or **cron_schedule**. A screenshot of **generated** proves local completion only; retain downstream acknowledgement separately when delivery is in scope.

Related

- [Generation Modes and Triggers](#)
- [Error Handling and Recovery](#)
- [Status Tracking](#)