magmodules

Cron Jobs

KatanaPIM for Magento 2

Technical documentation for the automated synchronization cron jobs. This guide explains how the cron jobs work together to keep your Magento catalog in sync with KatanaPIM.

Overview

KatanaPIM Connect uses two cron jobs that work together:

Cron Job	Default Schedule	Purpose
Full Update	Every 8 hours (`:33`)	Fetch all data from KatanaPIM API
Incremental Update	Every 5 minutes	Sync pending changes to Magento

This two-phase approach separates data fetching from Magento processing, allowing for efficient handling of large catalogs.

How They Work Together

Phase 1: Full Update (Data Retrieval)

The Full Update cron fetches ALL data from the KatanaPIM API but does **not** sync to Magento directly.

What it does:

1. Fetches all attributes, categories, products, and assets from KatanaPIM API
2. Stores the data in intermediate tables (`katanapim_product` , `katanapim_category` , etc.)
3. Calculates hashes to detect changes
4. Sets the `needs_update` flag on items where PIM data differs from Magento
5. Marks items no longer in PIM as deleted

What it does NOT do:

- Create or update Magento products
- Create or update Magento categories
- Import images into Magento

Phase 2: Incremental Update (Magento Sync)

The Incremental Update cron processes items marked for update and syncs them to Magento.

What it does:

1. Fetches recently changed items from KatanaPIM API (using timestamp filter)
2. Processes all items with `needs_update = 1`
3. Creates or updates Magento products, categories, and attributes
4. Imports images and videos into Magento
5. Clears the `needs_update` flag after successful sync

The needs_update Flag

This flag is the central mechanism that determines what gets synced to Magento. Understanding how it works is essential for troubleshooting sync issues.

Two Hash Comparisons

The module uses two separate hash comparisons:

1. During API fetch (Full Sync):

Compare	Action
New Katana hash = Stored Katana hash	Skip item, no changes in KatanaPIM
New Katana hash ≠ Stored Katana hash	Update stored data, proceed to step 2

2. When saving to intermediate table:

Compare	Result
Stored Katana hash = Stored Magento hash	`needs_update = 0`
Stored Katana hash ≠ Stored Magento hash	`needs_update = 1`

This means: if data hasn't changed in KatanaPIM since the last Full Sync, the item is skipped entirely and `needs_update` won't be updated.

How needs_update Gets Cleared

After a successful Magento sync:

1. The item is created or updated in Magento
2. `mapped_magento_hash` is set to match `mapped_katana_hash`
3. `needs_update` becomes `0`

Filtering Logic

Items are only synced to Magento when ALL conditions are met:

- `needs_update = 1`
- `skipped = 0`
- `is_deleted = 0`
- `has_error = 0`

Why needs_update Stays at 1

If products keep syncing repeatedly, it means the hashes don't match after sync. You can inspect the hash values in the **Katana PIM → Products** grid and product detail view.

Common causes:

Cause	Solution
Attribute mapping mismatch	Check all attributes are mapped in **Katana PIM → Attributes**
Select option mismatch	Ensure option labels match exactly (case-sensitive)
Missing store language mapping	Map all languages in **Stores → Configuration → Katana PIM → General → Store Languages**
Category not synced	Sync categories first before products
Related products missing	Ensure all related products exist in Magento

Excluded from hash comparison: `url_key`, `product_stock`, `product_price`, `video`, `website_prices`

Cron Limits

To prevent timeouts and memory issues, each cron run processes a limited number of items.

Where Limits Apply

Sync Type	API Fetch	Magento Sync
Full Sync (Cron)	No limit	Skipped (`only_api: true`)
Incremental Sync (Cron)	Uses timestamp filter	Limited per entity type
Full Sync (CLI)	No limit	No limit

The cron limits primarily affect the **Incremental Sync**, which processes items from the intermediate tables to Magento.

Default Limits (Incremental Sync)

Entity	Default Limit	Config Path
Products	100	`katanapim_automation/cron/max_products_per_cron`
Categories	100	`katanapim_automation/cron/max_categories_per_cron`
Attributes	100	`katanapim_automation/cron/max_attributes_per_cron`
Assets	100	`katanapim_automation/cron/max_assets_per_cron`

What This Means

If 500 products have `needs_update = 1`, but the limit is 100:

- First cron run: processes products 1-100
- Second cron run: processes products 101-200
- And so on...

With a 5-minute incremental schedule, all 500 products would be synced within ~25 minutes.

Cron Schedules

Default Schedules

Cron Job	Cron Expression	Meaning
Full Update	`33 */8 * * *`	At minute 33, every 8 hours
Incremental Update	`*/5 * * * *`	Every 5 minutes

Schedules can be configured in **Stores → Configuration → Katana PIM → Automation**.

Sync Order

Entities are processed in a specific order to maintain data integrity:

1. **Attributes** — Must exist before products can use them

2. **Categories** — Must exist before products can be assigned
3. **Products** — Main product data
4. **Assets** — Images and videos (requires products to exist)

Cron vs CLI

Aspect	Cron	CLI
Full sync behavior	API fetch only	API fetch + Magento sync
Output	Logged to files	Console output
Limits	Respects max per cron	No limits
Use case	Automated background sync	Manual imports, initial setup

Important: The CLI command `bin/magento katana:import:full` performs both phases (fetch + sync) in one run, while the cron full update only fetches data.

Troubleshooting

Items Not Syncing

Check the flags:

```
SELECT sku, needs_update, skipped, is_deleted, has_error
FROM katanapim_product
WHERE needs_update = 1
LIMIT 10;
```

Items need `needs_update = 1` AND `skipped = 0` AND `is_deleted = 0` AND `has_error = 0` to be processed.

Cron Not Running

Verify cron is active:

```
bin/magento cron:run --group=katanapim
```

Check cron schedule:

```
SELECT * FROM cron_schedule
WHERE job_code LIKE 'katanapim%'
ORDER BY scheduled_at DESC
LIMIT 20;
```

Sync Taking Too Long

Options:

1. Increase cron limits (see "Adjusting Limits" above)
2. Reduce incremental schedule frequency
3. Run full sync via CLI during off-peak hours instead of cron

Items Marked as Deleted

If items are incorrectly marked as deleted, it usually means:

- The item was removed from KatanaPIM
- API connection failed during full sync (partial fetch)

Fix: Run a manual full sync via CLI to re-fetch all data:

```
bin/magento katana:import:full --products
```

Sync Log

All cron runs are logged in **Katana PIM → Sync Log**.

Each entry shows:

- Sync type (Full/Incremental)
- Items created, updated, deleted
- Errors encountered
- Execution time

Log cleanup: Old sync logs are automatically removed after 30 days (configurable).

Need More Help?

Documentation:

- All Help Articles - Complete documentation overview

Support:

- Contact Support - Get help from our team

For a complete overview of features and configuration options, see the KatanaPIM extension on magmodules.eu

All articles for KatanaPIM

Installation

1	Installation using Composer (recommended)
2	Install through FTP and SSH

Configuration

1	Quick Start Guide
2	Configuration
3	Categories Grid
4	Attribute Mapping
5	Configurable Products Setup
6	Automation

Troubleshooting

1	Troubleshooting

Grids

1	Products Grid
2	Attributes Grid
3	Assets Grid
4	Sync Log Grid

Background

1	CLI Commands

2	**Cron Jobs (current)**
3	Product to Website Matching
4	Understanding the Sync Process