CLI Commando's

KatanaPIM for Magento 2

Command line tools for the KatanaPIM Connect extension. These commands are useful for manual synchronization, automation scripts, and troubleshooting.

Available Commands

Command	Description
`katana:import:full`	Run full synchronization from KatanaPIM
`katana:import:incremental`	Run incremental synchronization (delta)
`katana:assets:cleanup`	Remove soft-deleted assets from Magento media gallery
`katana:migrate:attributes`	Migrate attribute mappings from Itonomy_Katanapim module

katana:import:full

Performs a complete synchronization of all entities from KatanaPIM.

```
# Sync everything (attributes, categories, products, assets)
bin/magento katana:import:full

# Sync only attributes
bin/magento katana:import:full --attributes

# Sync only categories
bin/magento katana:import:full --categories

# Sync only products
bin/magento katana:import:full --products

# Sync only assets
bin/magento katana:import:full --assets
```

Options:

Option	Description
`--attributes`	Run attributes update only
`--categories`	Run categories update only
`--products`	Run products update only

Option	Description
`--assets`	Run assets update only
`--force`, `-f`	Force update all products, ignoring hash check
`--katana-pim-id=<id>`	Scope the entire run to one KatanaPIM product

When no options are provided, all entity types are synchronized in the correct order.

Force Update: *(available from v3.7.0)*

Use the `--force` flag when you need to re-sync products regardless of whether the data has changed:

```
# Force re-sync all products (ignore hash comparison)
bin/magento katana:import:full --products --force

# Short version
bin/magento katana:import:full --products -f
```

This is useful when:

- Magento data was modified outside of the sync process
- Hash comparison isn't detecting changes correctly
- You need to overwrite local changes with PIM data

Single Product Sync: *(available from v3.11.0)*

Use `--katana-pim-id` to sync exactly one product instead of everything that is pending:

```
# Sync one product end to end
bin/magento katana:import:full --katana-pim-id=1234

# Combine with --force to ignore the hash comparison
bin/magento katana:import:full --katana-pim-id=1234 --force

# Skip the API and only push what is already stored locally
bin/magento katana:import:full --katana-pim-id=1234 --skip-api
```

The scope applies to the whole run:

- The API fetch requests that single product (and its assets) instead of paging through everything
- The Magento product and asset sync are limited to that product
- Only the `products` and `assets` entities run. Add `--attributes` or `--categories` explicitly if the product needs a refreshed attribute or category mapping

- Reindexing is limited to that product's Magento entity instead of a full `reindexAll()`

Nothing is ever marked as deleted during a scoped run. A normal full sync marks everything the API did not return as deleted, which is only safe when the run actually saw the complete catalog. A scoped run sees one product, so that step is skipped and the output states `Skipped marking deleted (scoped run)`.

Use this when:

- Debugging why one specific product will not sync
- A support question about a single SKU
- Verifying a fix without waiting for a full run

How sync works:

The synchronization operates in two phases:

1. **Data Retrieval** — Fetches current data from the KatanaPIM API
2. **Magento Updates** — Creates or updates entities based on your configuration

Whether new entities are automatically created depends on your settings at **Stores → Configuration → KatanaPIM → Automation**:

- Auto create new categories
- Auto create new products

Output: Shows progress and results for each entity type.

Use this when:

- Initial setup and first import
- After major PIM changes
- Weekly/daily maintenance
- Troubleshooting sync issues

katana:import:incremental

Performs a delta synchronization, importing only items that have changed since the last sync.

```
bin/magento katana:import:incremental
```

Output: Shows progress and number of items updated.

Use this when:

- Regular scheduled updates
- Keeping Magento in sync throughout the day
- After individual product changes in PIM

katana:assets:cleanup

Removes soft-deleted assets (marked as `is_deleted=1`) from the Magento media gallery. Assets are soft-deleted when they are removed in KatanaPIM but the actual media files remain in Magento. This command cleans up those orphaned gallery entries.

(Available from v3.8.0)

```
# Show what would be deleted without making changes
bin/magento katana:assets:cleanup --dry-run

# Remove soft-deleted assets from media gallery
bin/magento katana:assets:cleanup
```

Options:

Option	Description
`--dry-run`	Show what would be deleted without actually deleting

Use this when:

- After removing images from products in KatanaPIM
- To clean up orphaned media gallery entries
- As part of periodic maintenance

katana:migrate:attributes

Migrates attribute mappings from the old `Itonomy_Katanapim` module to `KatanaPim_Connect`. This command is only needed when upgrading from the legacy Itonomy module.

(Available from v3.10.0)

```
# Preview what would be migrated
bin/magento katana:migrate:attributes --dry-run

# Run the migration
bin/magento katana:migrate:attributes
```

Options:

Option	Description
`--dry-run`	Show what would be migrated without making changes

How it works:

1. Reads attribute mappings from the old `katanapim_attribute_mapping` table
2. Finds matching attributes in the new `katanapim_attribute` table by `katana_attribute_code`
3. Copies the `magento_attribute_code` and `is_configurable` settings to the new table

Prerequisites:

- The old `katanapim_attribute_mapping` table must exist (from a previous Itonomy_Katanapim installation)
- Run a full API attribute import first so the new table has entries to match against

Use this when:

- Migrating from the Itonomy_Katanapim module to KatanaPim_Connect
- You want to preserve existing attribute mappings instead of reconfiguring manually

Examples

Initial Setup

```
# First time setup - import everything
bin/magento katana:import:full
```

Daily Maintenance

```
# Full sync during maintenance window
bin/magento cache:flush
bin/magento katana:import:full
bin/magento indexer:reindex
```

Incremental Updates

```
# Quick sync of recent changes
bin/magento katana:import:incremental
```

Step-by-Step Import

For large catalogs, import in stages:

```
# Step 1: Import and map attributes first
bin/magento katana:import:full --attributes

# Step 2: Configure attribute mapping in admin
# (manual step - use Attributes grid)

# Step 3: Import categories
bin/magento katana:import:full --categories

# Step 4: Import products
bin/magento katana:import:full --products

# Step 5: Import images and videos
bin/magento katana:import:full --assets
```

Troubleshooting

```
# Run with verbose output
bin/magento katana:import:full -v

# Check what would be synced without making changes
bin/magento katana:import:full --attributes
```

Indexer Recommendations

For large catalogs, set your indexers to "Update on Schedule" mode before running imports:

```
# Check current indexer modes
bin/magento indexer:show-mode

# Set all indexers to schedule mode
bin/magento indexer:set-mode schedule
```

Why "Update on Schedule"?

With "Update on Save" mode, Magento triggers reindexing after every single entity import — extremely slow for substantial datasets. Schedule mode queues index updates and processes them in batches, significantly

improving import performance.

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	Installatie met Composer (aanbevolen)
2	Installatie via FTP en SSH

Configuration

1	Snelstartgids
2	Configuratie
3	Categorieën Grid
4	Attribuut Mapping
5	Configureerbare Producten Instellen
6	Automatisering

Troubleshooting

1	Probleemoplossing

Grids

1	Producten Grid
2	Attributen Grid
3	Assets Grid
4	Sync Log Grid

Background

1	**CLI Commando's (current)**

2	Cron Jobs
3	Product to Website Matching
4	Het sync proces begrijpen

