

Fast Asynchronous Re-indexing 1.1.9

Getting Started

Welcome to the **Fast Asynchronous Re-indexing Documentation**. Whether you are new or an advanced user, you can find useful information here. **Next steps:**

- How to install extension
- Configuration Settings

How to install extension

1. Login to Magento backend.
2. Go to **System > Tools > Compilations**. If Compiler Status is **Enabled**, disable the compilation by clicking the button **Disable** in the upper right corner.
3. Backup your store database and web directory.
4. Download extension from your Personal Account.
5. Unzip the extension locally.
6. Copy contents of the folder **step1** to the root directory of Magento store.
7. Copy contents of the folder **step2** to the root directory of Magento store.
8. Completely clear the site cache. Go to **System > Cache Management**. Click button **Flush Magento Cache**. If you want to clear the cache manually, you need to remove a folder `/var/cache` at the server.
9. Logout from Magento backend and login again.
10. Go to **System > Configuration > Mirasvit Extensions > System**, at form **Installed Extensions** press the button **Run validation tests for all extensions** to check extension installation and configuration.
11. If compilation was enabled before installation, you need to run recompilation process to enable compilation. Go to **System > Tools > Compilation** and click on **Run Compilation Process**. If you did not disable compilation and store is down after installation, follow the link and make next steps to recover the store.

Next steps:

- How to Setup Cron for Magento
- Configuration settings
- Enabling extension

How to upgrade extension

To upgrade extension follow next steps:

1. Login to Magento backend.
2. Go to **System > Tools > Compilations**. If Compiler Status is **Enabled**, disable the compilation by clicking the button **Disable** in the upper right corner.
3. Backup your store database and web directory. If you have customizations, please, create a separate backup of **app/code/local/Mirasvit** before upgrading to the new version.
4. Download new extension package from your Personal Account.
5. Unzip the extension locally.
6. Temporary disable extension. Check Temporary Disabling.
7. Copy contents of the folder **step1** to the root directory of Magento store.
8. Copy contents of the folder **step2** to the root directory of Magento store.
9. Completely clear the site cache. Go to **System > Cache Management**. Click button **Flush Magento Cache**. If you want to clear the cache manually, you need to remove a folder **/var/cache** at the server.
10. Logout from Magento backend and login again.
11. If compilation was enabled before upgrade, you need to run recompilation process to enable compilation. Go to **System > Tools > Compilation** and click on **Run Compilation Process**. If you did not disable compilation and store is down after upgrade, follow the link and make next steps to recover the store.

Temporary Disabling

To temporary disable extension please follow the next steps:

- Disable Compilation Mode (if enabled).
- Remove file **app/etc/modules/Mirasvit_AsyncIndex.xml**.
- If you have only one extension from Mirasvit installed, remove file **app/etc/modules/Mirasvit_MstCore.xml**.
- Login in to Magento back-end and refresh site cache (if enabled).

Extension Removing

To uninstall extension please follow the next steps:

- Disable Compilation Mode (if enabled).
- Remove file **app/etc/modules/Mirasvit_AsyncIndex.xml**.
- If you have only one extension from Mirasvit installed, remove file **app/etc/modules/Mirasvit_MstCore.xml**.
- Login in to Magento back-end and refresh site cache (if enabled).
- Remove folder **app/code/local/Mirasvit/AsyncIndex**.

How to Setup Cron for Magento

General settings

Go to **System > Configuration > Mirasvit Extensions > Fast Asynchronous Re-indexing**.

Description of the fields:

- **Asynchronous full reindex** - if option enabled, full reindex will be processed in background by cron. I.e. after select **Reindex Data** at **Index Management**, extension will add selected indexes to queue and then process this queue via cron.
- **Asynchronous reindex for changes** - if option enabled, reindex of changed items will be processed in background by cron.
- **Processing Mode** - the mode re-indexing queue:
 - **Per Entity** - extension will process queue one by one.
 - **Per Index** - extension will process queue index by index.

For example, we have in queue few items: **Product A, Product B, Product C**.

- If **Processing Mode** is **Per Entity** extension will process queue by products: **Product A**
Product Attributes
Product Prices
Catalog URL Rewrites
Product Flat Data
Category Flat Data
Category Products
Search Index
Stock Status
Tag Aggregation Data
Product B
Product Attributes
Product Prices
Catalog URL Rewrites
Product Flat Data
Category Flat Data
Category Products
Search Index
Stock Status
Tag Aggregation Data
Product C
Product Attributes
Product Prices
Catalog URL Rewrites

Product Flat Data
Category Flat Data
Category Products
Search Index
Stock Status
Tag Aggregation Data

- If **Processing Mode** is **Per Index** extension will process queue by indexes: **Product Attributes**

Product A

Product B

Product C

Product Prices

Product A

Product B

Product C

Catalog URL Rewrites

Product A

Product B

Product C

Product Flat Data

Product A

Product B

Product C

Category Flat Data

Product A

Product B

Product C

Category Products

Product A

Product B

Product C

Search Index

Product A

Product B

Product C

Stock Status

Product A

Product B

Product C

Tag Aggregation Data

Product A

Product B

Product C

- **Validate Product Index** - if option enabled, extension will validate products index. If index for some product is incorrect, extension will add this product to the reindexing queue. Usually this

feature should be enabled if you import products from external system directly to the database (e.g. use Magmi).

The option **Use Flat Catalog Product** (System > Configuration > Catalog - Frontend) must be enabled for correct work of this feature.

- **Validate Category Index** - if option enabled, extension will validate categories index. If index for some category is incorrect, extension will add this category to the reindexing queue. Usually this feature should be enabled if you import categories from external system directly to the database (e.g. use Magmi).

The option **Use Flat Catalog Category** (System > Configuration > Catalog - Frontend) must be enabled for correct work of this feature.

- **Max queue batch size for processing** - sets how many products/categories can be added to reindex queue (if product or category index incorrect).
~!This option available, only if option **Validate Product index** is enabled. This variable depends on your server. Recommended value 100 for slower server and < 1000 for fast server.
- **Reindex by cron** - if option enabled, extension will run reindexing of queue via cron.
By default extension use default magento cron **cron.php**.
Also, you can disable this option and create custom cronjob which should run the file via Command Line Interface.
- **Ignored indexes** - list of ignored indexes. Selected indexes will not be processed (reindexed) by our extension. They can use default magento reindexing mechanism. Click **Save Config** to save all the changes.

Enabling extension

For enable extension, you need configure our extension.

1. Go to **System > Index Management**.
2. Point all the indexes with the **Select All** link.
3. In the **Actions** dropdown list, select **Change Index Mode**.
4. In the dropdown list **Index Mode** that appears, chose the **Manual Update** and press to **Submit**.
You will see success message "Total of 9 index(es) have changed index mode."

✔ Total of 9 index(es) have changed index mode.

Force process queue

Detailed Log

Reindexing Completed (last run 1 min 18 sec ago)

Index Management

Select All Unselect All Select Visible Unselect Visible 9 items selected					Actions	Change Index Mode
	Index	Description	Mode	Status	Update Required	
✔	Product Attributes	Index product attributes for layered navigation building	Manual Update	READY	NO	
✔	Product Prices	Index product prices	Manual Update	READY	NO	
✔	Catalog URL Rewrites	Index product and categories URL rewrites	Manual Update	READY	NO	
✔	Product Flat Data	Reorganize EAV product structure to flat structure	Manual Update	READY	NO	
✔	Category Flat Data	Reorganize EAV category structure to flat structure	Manual Update	READY	NO	
✔	Category Products	Indexed category/products association	Manual Update	READY	NO	
✔	Search Index	Rebuild search index (Products, Categories, 1)	Manual Update	READY	NO	
✔	Stock Status	Index Product Stock Status	Manual Update	READY	NO	
✔	Tag Aggregation Data	Rebuild Tag aggregation data	Manual Update	READY	NO	

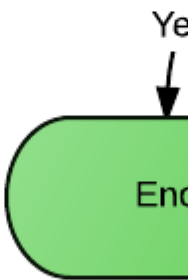
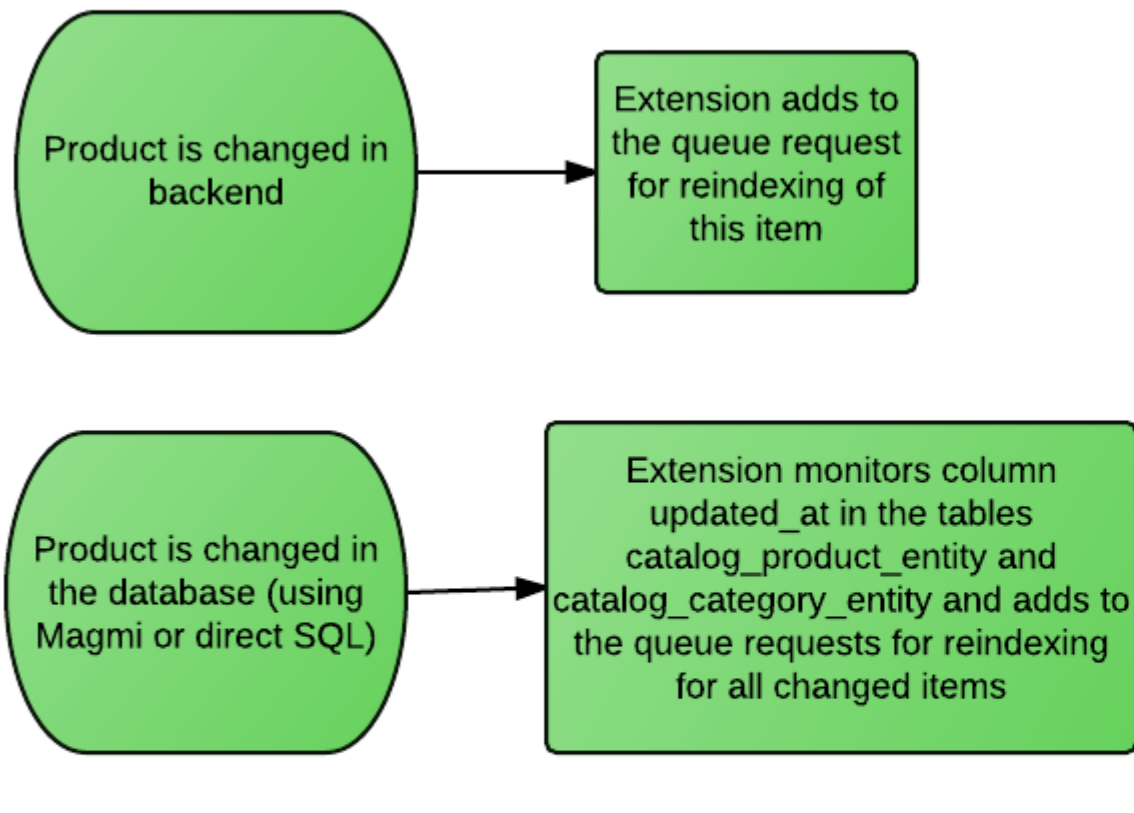
If you use **Magento Enterprise Edition 1.13 or higher**, please also follow these steps:

1. Go to **System > Configuration > Index Management**.
2. Change index mode for all indexes to **Update when scheduled**.
3. Press **Save Config**.

As result, you will see additional Event/Action in reindex queue - **Enterprise Index Refresh**.

How extension works?

Fast Asynchronous Re-indexing



Command Line Interface

To run extension via command line (shell, SSH) you can use php script located at **/shell/asyncindex.php**. Usage: `php asyncindex.php [options]` Run (reindex queue, index validation (if enabled)) `--fill-queue` Generate random queue `--help` Help Examples `php asyncindex.php php asyncindex.php --fill-queue` Additionally, for handle indexes you can use native magento php script located at **/shell/indexer.php**. Usage: `php indexer.php [options]` `--status <indexer>` Show Indexer(s) Status `--mode <indexer>` Show

Indexer(s) Index Mode --mode-realtime <indexer> Set index mode type "Update on Save" --mode-manual <indexer> Set index mode type "Manual Update" --reindex <indexer> Reindex Data info Show allowed indexers reindexall Reindex Data by all indexers help This help <indexer> - Comma separated indexer codes or value "all" for all indexers (if without codes). To see all available **indexers**, run php indexer.php info Examples php indexer.php reindexall php indexer.php --reindex catalog_product_attribute

Troubleshooting

I go to the System > Index Management, click the button Detailed Log and see red error message(s)

This problem happens if our extension tries to reindex some record and reindexing process fails with error. This error can be caused by many factors (e.g. error in 3rd party extension). This error is not connected to our extension. **Solution:**

1. Disable our extension
2. Run full reindex via SSH: # go to the magento shell folder php -f indexer.php reindexall
3. Fix **ALL** errors that you see during full reindex
4. Enable our extension

Frequently Asked Questions

1.

What Processing Mode should I use: Per Entity or Per Index?

Per Entity mode process reindex queue per product, while **Per Index** mode process queue per products index.

For more information about Processing Modes, check the General Settings page

2.

When i use the Processing Mode - Each Index, does it do a full reindex, or only indexes changes?

Each Index reindex only changes too (all changes for first index, for second index etc)

3.

What Processing Mode makes reindex faster?

Per Index mode works faster, but it less reliable, because it requires more memory for executing. As a result, there is a possibility of reaching memory limits and locking the reindex process.

4.

What is a purpose of the Force process queue button?

Force process queue allows to make reindex via web browser.

If after clicking the button reindex doesn't start, it means that reindex currently executing by cron.

5.

How to test the extension works correctly or not?

Go to **System > Index Management**. Click the button **Detail Log** and check the lines with the **State** status **Error**.

Other statuses indicate that the extension works correctly.

6.

What is the relationship between the time of reindexing, the product amount, the number of store view, server performance?

Time of reindexing depends on the number of products and storeviews. The more of them exist, the more time need to make reindex.

Server capacity also effects on the time of reindex process.

7.

How to know what is the most appropriate value for "Max queue batch size for processing" option?

This variable depends on your server capacity. Recommended value for slower servers is 100 and for fast servers < 1000.

You can set the approximate value and check if the reindex was successfully finished.

If reindex process wasn't executed, you can reduce value until it fits your server configuration.

8.

How to set up extension for various import schemes?

- Default import mechanism.
Doesn't require additional settings.
- Import through Magmi
Extension settings **Validate Product Index** and **Validate Category Index** must be enabled for successful import.
- Import through a user script
Before importing script, you need to disable **Asynchronous full reindex** option at **General Settings**.
After script was imported, make native magento reindex via SSH (command of native magento reindex described at Command Line Interface page).

9.

What is the difference between Full Reindex and Reindex for changes?

Full Reindex performs reindex of all products, while **Reindex for changes** makes reindex of only changed products.

Usually, you should reindex only changed products and perform full reindex only in critical situations.

10.

Does Full reindex can run through default mechanism of Magento, but run reindex of the changed products via extension?

Yes, you can set up such conditions. Go to **System > Configuration > Mirasvit Extensions > Fast Asynchronous Re-indexing**.

Set **No** for the **Asynchronous full reindex** and set **Yes** for **Asynchronous reindex for changes** option.

11.

When I run indexing through asynchronous reindex why CPU load is high?

The same CPU load you can see when you run reindex without our extension.

Extension improve time of reindex by moving all reindex tasks to cron.

12.

In which cases should be used "Clear the queue" button?

This option can be used when some problems are appeared with extension work.

For example, when a lot of products were added to reindex queue and extension can't handle with such amount of products.

Another possible reason, when extension makes reindex more slowly then before.

13.

What steps should i do to clear the queue successfully?

Follow next steps:

- Disable the extension module
- Delete files on your server at **/var/lock/**
- Clear the **index_event** table.

Change Log

1.1.13

(2016-11-24) Fixed

- Fixed an issue with Enterprise index observer

1.1.11

(2016-11-22) Fixed

- Fixed a possible issue with product validator

1.1.10

(2016-11-01) Improvements

- Improved cache clearing logic

Documentation

- Added FAQ

1.1.9

(2016-02-29) Fixed

- Fixed an issue with "Updated At" time for indexes

1.1.8

(2016-02-22) Improvements

- Improved performance for cache clear task (per entity mode)

1.1.7

(2016-02-15) Fixed

- Minor bug fixes related with detailed log

1.1.6.1

(2015-04-12) Fixed

- Fixed an issue with version comparison for product flat helper
- Fixed an issue with serialization of error messag
- Fixed an issue with reindex search index (EE SOLR)

Improvements

- Improved user interface

1.1.6

(2013-12-09) Fixed

- Fixed an issue with catalog inventory reindex
- Fixed an issue with loop for event catalog_reindex_price

Improvements

- Added ability to ignore indexes during reindex
- Implemented 2 reindex modes: per entity, per index
- Added detailed log to Index Management page