

Advanced Sphinx Search Pro 2.3.2

Getting Started

Welcome to the **Advanced Sphinx Search Pro Documentation**. Whether you are new or an advanced user, you can find useful information here. First of all we recommend to check the following links:

- How to install extension
- Configure Search Results

How to install extension

1. Login to Magento backend.
2. Go to **System > Index Management**. Run **Full Reindex** of all indexes. If it's not possible to run reindex via browser, try to run it via SSH. If you see errors, fix them now.
3. Go to **System > Tools > Compilations**. If Compiler Status is **Enabled**, disable the compilation by clicking the button **Disable** in the upper right corner.
4. Backup your store database and web directory.
5. Download extension from your Personal Account.
6. Unzip the extension locally.
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. Go to **Search > Manage Search Indexes** and configure Product index.
12. Go to **System > Index Management**. Run **reindex** of the **Search Index**.
13. Go to **System > Configuration > Mirasvit Extensions > System**, at form **Installed Extensions** press the button **Run validation tests for all extensions** for check extension installation and configuration.
14. 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.

Our extension can not work properly, if there installed other third-party search extensions.

If you have installed such an extensions, please, **turn them off** either before installation, or immediately after installation at `/app/etc/modules` folder, then purge ALL cache, that you're using, and run reindex.

If you're using Enterprise Edition, you also need to disable **Enterprise_Search** and **Enterprise_CatalogSearch**. **Next step:**
Configure Search Results

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.
12. If necessary adjust css styles to fit your store's frontend theme.

Extension Disabling

Temporary Disabling

To temporary disable extension please follow the next steps:

- Disable Compilation Mode (if enabled).
- Remove following files:
 - **app/etc/modules/Mirasvit_SearchIndex.xml**
 - **app/etc/modules/Mirasvit_SearchSphinx.xml**
 - **app/etc/modules/Mirasvit_SearchLandingPage.xml**
- If you have only one extension from Mirasvit installed, remove following 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 following files:
 - app/etc/modules/Mirasvit_SearchIndex.xml
 - app/etc/modules/Mirasvit_SearchSphinx.xml
 - app/etc/modules/Mirasvit_SearchLandingPage.xml
- If you have only one extension from Mirasvit installed, remove following file:
 - app/etc/modules/Mirasvit_MstCore.xml
- Login in to Magento back-end and refresh site cache (if enabled).
- Remove following folders:
 - app/code/local/Mirasvit/SearchIndex
 - app/code/local/Mirasvit/SearchSphinx
 - app/code/local/Mirasvit/SearchLandingPage
- Remove templates and layouts files of extension from your theme folder. List of such files can be collected from files in the extension package.

Configure a global search options

Go to **System > Configuration > Mirasvit Extensions > Sphinx Search**. You'll see following options:

- **Max number of items in the result** - sets maximum number of items in the search result. Set **0** to disable limitation. If you have a large catalog, we suggest to limit the number of search results to 1000.
- **Min relevance value to display** - sets minimal relevance of items in the search result. Set **0** to disable limitation. It means if some of the products have less search relevance value than set there, the product will not be displayed.
- **Match Mode** - sets search logic for the search queries. There are following match modes available:
 - **AND** - search will be using AND logic. Search result must be matching all words from the search query.
 - **OR** - search will be using OR logic. Search result must be matching any word from the search query.
- **Enable wildcard search** - allows customer to search the product by part of the word. This option slightly reduces the relevance of search and increases the number of search results. There are 4 options enabled:
 - **Enabled (*word*)** - allows to find the product, if the search phrase is contained in the beginning, end or in the middle of the product. Example: if the search query is "**phon**", search results can be: "**telephone**", **phone**, **aphon**.
 - **Enabled at end (word*)** - allows to find products, which only start by search phrase. Example: if the search query is "**phon**", search results can be: "**phone**" or "**phonetic book**", but not: "**telephone**" or **iphone**.

- **Enabled at start** (*word) - allows to find products, which only end by search phrase. Example: if the search query is "**phon**", search results can be: **aphon** or **typhon**, but not: "telephone" or **phone**.
 - **Disabled** - all search results will be displayed exactly as search query. Example: if the search query is "**phon**", search result can be only for "**phon**", but not for "**phone**", "tele**phon**e" or a "**phon**".
- **Wildcard Exceptions** - the list of words (characters) for which no wildcard rules will be applied. If **wildcard search** is enabled, search query 'men shoes', will also return products with "women" word. If we add 'men' to exceptions list, search will return only 'men' results.
- **Replace words in search query** - words from the first column will be replaced by word from the second column. Words in first column to be separated by comma. You have a record 'notebook, computer' => 'laptop'. Search phrases 'asus notebook' and 'asus computer' will be modified to 'asus laptop'. Search will be done by modified phrase. Though the search is performed by the replaced query, the original query, entered into the search box, remains unchanged.
- **'Not' words** - words which change search sequence to '**NOT**' (ex. not, without, exclude). 'Not' words can be useful for search phrases like 'laptops without bluray'. In this case search will return all laptops without word 'blu-ray'.
- **Redirect if Single Result** - if the search query results only one match, customer will be redirected to the single result automatically.
- **Enable Google search snippets** - if option enabled, extension shows Sitelink Search Box on the Google search results page. After option enabling, search box will be shown only after Google reindexing. Sitelink Search Box on the Google search results page



youtube

Web

Videos

News

Shopping

Images

More ▾

Search tools

About 1,700,000,000 results (0.22 seconds)

YouTube

www.youtube.com/ ▾

Hosts user-generated videos. Includes network and professional content.

guilty dog

guilty dog

guilty dog video

denver guilty dog



when

YouTube Subscriptions

Share your videos with friends, family, and the world.

Movies

YouTube Movies (United States).

You Tube

TRAYAURUS TAKES OVER
YOUTUBE | Minecraft. by ...

Education

Whether you're doing research for a project, need help with ...

- **Display Related Search Terms** - if option enabled, related search terms will be displayed on the search result page. **Example:**

SHOP BY

SHOPPING OPTIONS

→ CATEGORY

[Electronics](#) (4)

→ PRICE

[\\$0.00 - \\$999.99](#) (3)

[\\$2,000.00 and above](#) (2)

→ COLOR

[Black](#) (4)

[Silver](#) (1)

→ BRAND

[Sony](#) (2)

Did you know?

Our customer service is available **24/7**

(555) 555-0123

Search results for 'son'

Products (5)

Attribute: Brand (1)

Related Searches: [sony](#), [Sony VAIO](#), [sony notebook](#)

5 Item(s)

Show 9 per page

View as: [Grid](#) [List](#)

Sort By [Relevance](#) ↓



Sony VAIO 11.1"
Notebook PC

As low as: **\$2,699.99**



Sony VAIO
VGN-TXN27N/B 11.1"
Notebook PC

★★★★★ [10 Review\(s\)](#)

\$2,699.99



Sony Ericsson W810i

★★★★★ [3 Review\(s\)](#)

\$399.99

- **Enable search terms highlighting** - if option enabled, search query word(s) will be highlighted in the search results. **Example:**

SHOP BY

SHOPPING OPTIONS

→ CATEGORY

[Electronics](#) (4)

→ PRICE

[\\$0.00 - \\$999.99](#) (3)

[\\$2,000.00 and above](#) (2)

→ COLOR

[Black](#) (4)

[Silver](#) (1)

→ BRAND

[Sony](#) (2)

Did you know?

Our customer service is
available **24/7**

(555) 555-0123

Search results for 'son'

Products (5)

Attribute: Brand (1)

5 Item(s)

Show per page

View as: **Grid** [List](#)

Sort By **Relevance** ↓



Sony VAIO 11.1"
Notebook PC

As low as: **\$2,699.99**



Sony VAIO
VGN-TXN27N/B 11.1"
Notebook PC

★★★★☆ [10 Review\(s\)](#)

\$2,699.99



Sony Ericsson W810i

★★★★☆ [3 Review\(s\)](#)

\$399.99

- **"Long Tail" search** - allows to receive correct search results for words that contain dashes or any other non-alphabetic symbols. For more information, see "Long Tail" search configuration
- **Display multi-store results** - if option enabled, search results will be displayed for each store in the different tabs. To set titles of the stores in the search result tabs, go to **Search>Manage Search Indexes**, open **Product index** and set **Title** for each store view. **Example of the multi-store result tabs:**

Home / Search results for: 'samsung'

SHOP BY

SHOPPING OPTIONS

CATEGORY
[Electronics \(2\)](#)

CONTRAST RATIO
[3000:1 \(1\)](#)

PRICE
[\\$100.00 - \\$199.99 \(1\)](#)
[\\$300.00 and above \(1\)](#)

COLOR
[Silver \(1\)](#)



MANUFACTURER
[Samsung \(1\)](#)

Products (2) **English Store (2)** **French Store (2)**

Search results for 'samsung'

2 Item(s) Show 9 per page

View as: [Grid](#) [List](#) Sort By [Relevance](#) ↓

 <p>Samsung MM-A900M Ace ★★★★★ 2 Review(s) \$150.00</p>	 <p>22" Syncmaster LCD Monitor \$399.99</p>
--	---

- **Display results for these stores** - sets stores, for which search results will be displayed.
- **Redirect to the store with results** - if the search results of the current store don't contain any product, a customer will be redirected to the search results of the another store view, where the search results exist.
- **Enable redirect from 404 to search results** - when customer proceeds to an old or wrong store product URL, he usually redirects to "404 Not Found" page. If option enabled, customer will be redirected to the store search results of the broken URL text, instead of "404 Not Found" page. For example, customer goes to the URL: <http://example.com/samsung/s500.html>. If the store page doesn't exist, customer will be redirected to the store search results page with search query words: samsung and s500: <http://example.com/catalogsearch/result/?q=samsung+s500>

Install a Sphinx Engine

If you would like to use our extension with Sphinx Engine, you need to setup it. To do it run following commands via SSH as root or using sudo: `wget http://sphinxsearch.com/files/sphinx-2.2.6-release.tar.gz`
`tar xvf sphinx-2.2.6-release.tar.gz`
`cd sphinx-2.2.6-release`
`./configure`
`make`
`make install`
Configuration file will be generated automatically. Extension reindex Sphinx Engine indexes by default magento cronjob. Check How to set up CRON Job for Magento.

How to install Sphinx Engine with Stemmer (language specific morphology)

Libstemmer supports morphology of following languages:

- danish
- dutch
- english
- finnish
- french
- german
- hungarian
- italian
- norwegian
- porter
- portuguese
- romanian
- russian
- spanish
- swedish
- turkish

To install Sphinx Engine with libstemmer run following commands via SSH as root or using sudo: `wget http://sphinxsearch.com/files/sphinx-2.2.6-release.tar.gz tar xvf sphinx-2.2.6-release.tar.gz cd sphinx-2.2.6-release wget http://snowball.tartarus.org/dist/libstemmer_c.tgz tar xvf libstemmer_c.tgz ./configure --with-libstemmer make make install` Then in file `app/code/local/Mirasvit/SearchSphinx/etc/conf/section.conf` add codes of additional languages to variable `morphology`. List of codes for different languages http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes. To add German to list of supported languages, you should have `morphology = stem_enru, libstemmer_de` in `section.conf`. **Run full reindex of Sphinx Engine indexes after changes.**

How configure charset tables?

`charset_table` is the main workhorse of Sphinx tokenizing process, ie. the process of extracting keywords from document text or query text. It controls what characters are accepted as valid and what are not, and how the accepted characters should be transformed (eg. should the case be removed or not). If charset table is configured for special language chars search phrases `kottkvarn` and `köttkvarn` will return the same result. To configure `charset_table`, you need in the file `app/code/local/Mirasvit/SearchSphinx/etc/conf/section.conf` add charset table for variable `charset_table`. List of codes for different charset tables http://sphinxsearch.com/wiki/doku.php?id=charset_tables **Next Step:**

- Connect the extension with Sphinx Engine

How to set up CRON Job for Magento

Setting up the cronjob using shell access

1. Before setting up a new cron job, check that it has not been set up yet. Use following command to get a list of cron jobs: `crontab -l`
2. To add a new cron task, open crontab for editing: `crontab -e`
3. Add the following record to the end of the list of jobs: `* * * * * /[path_to_php]/php -f /[path_to_magento]/cron.php >> /[path_to_magento]/var/log/cron.log 2>&1`

Setting up the cronjob using control panel (CPanel, Plesk, etc)

1. Open settings of cron jobs in your control panel.
2. Before setting up a new cron job, check that it has not been set up yet.
3. Add a new cron job with the following parameters:
 - Minute: *
 - Hour: *
 - Day: *
 - Month: *
 - Weekday: *
 - Command: `/[path_to_php]/php -f /[path_to_magento]/cron.php >> /[path_to_magento]/var/log/cron.log 2>&1`
4. Click the button **Click Add New Cron Job**.

Connect with Sphinx Engine

Make sure that you have installed a Sphinx Engine. See [How to Install a Sphinx Engine](#). Go to **System > Configuration > Mirasvit Extensions > Sphinx Search**. In dropdown **Search Engine** select **External Search Engine**. Fill in following fields:

- **Sphinx Host** - sphinx daemon host (localhost by default).
- **Sphinx Port** - sphinx daemon port (any free port).
- **Sphinx Bin Path** - if extension can't find "searchd" in your server, you need to enter the full path to the "searchd" (ex. `/usr/local/bin/`).
- **Cron expression for Full Reindex Schedule** - specifies how often cron must run full Sphinx reindex. For example, `0 3 * * *` - every day at 3:00.
- **Cron expression for Delta Reindex Schedule** - specifies how often cron must run delta (only changes) Sphinx reindex. For example, `*/15 * * * *` - every 15 minutes.
- **Base Path** - any folder on external server, where you would like to store sphinx configuration and indexes (ex. `/var/sphinx/`).

Connect with Sphinx Engine installed on other

server

For make extension work with Sphinx Engine installed on **other server** or installed on **Windows based server**, follow these steps: Before you generate a Sphinx configuration file, you must ensure that you have correct settings for Sphinx Engine connection to the database. Go to [magento_root]/app/etc/ and edit **local.xml** file. In the line <host> set IP address of the database server: <default_setup><connection> <host>85.75.44.153</host> ~! Don't use **localhost** or **127.0.0.1** as server IP address. You need to change host IP address even if your Magento and database are placed on the same server.

1. Go to **System > Configuration > Mirasvit Extensions > Sphinx Search**.
 2. Fill in following fields:
 - o **Base Path** - any folder on sphinx server, where you would like to store sphinx configuration and indexes (ex. /var/sphinx/)
 - o **Sphinx Host** - sphinx daemon host
 - o **Sphinx Port** - sphinx daemon port
 3. Press button **Save Config**.
 4. Press button **Generate Config file**.
 5. Copy generated config file to the **Base Path** on sphinx server.
 6. On sphinx server, run following commands:
 - o `indexer --config <base_path>/sphinx.conf --rotate mage_catalog_product`. Where `mage_catalog_product` code of index. If you use few indexes (products, categories, cms pages etc), you need do this command for each index.
 - o `searchd --config <base_path>/sphinx.conf`
 7. If configuration is correct and store server can connect to sphinx daemon, you will see search queries in sphinx log file (<base_path>/searchd_query.log)
 8. After done key configuration, you should configure cronjob for automatically update sphinx indexes.
 - o Full reindex `indexer --config <base_path>/sphinx.conf --rotate mage_catalog_product`
 - o Delta reindex `indexer --config <base_path>/sphinx.conf --rotate delta_mage_catalog_product` `indexer --config <base_path>/sphinx.conf --merge mage_catalog_product delta_mage_catalog_product --merge-dst-range deleted 0 0 --rotate`
 - o Start sphinx daemon `searchd --config <base_path>/sphinx.conf`
 - o Stop sphinx daemon `killall -9 searchd`
 - o Restart sphinx daemon `killall -9 searchd; searchd --config <base_path>/sphinx.conf;`
- Full example of crontab**
- ```
30 * * * * indexer --config sphinx.conf --rotate delta_mage_catalog_product; indexer --config sphinx.conf --merge mage_catalog_product delta_mage_catalog_product --merge-dst-range deleted 0 0 --rotate; killall -9 searchd; searchd --config sphinx.conf; 0 2 * * * indexer --config sphinx.conf --rotate mage_catalog_product; killall -9 searchd; searchd --config sphinx.conf;
```
- First job run delta reindex each hour (00:30, 01:30, 02:30 etc) and restart sphinx after delta reindex. Second job run full reindex every day at 02:00 and restart sphinx daemon. If you have few search indexes, you should run indexer command for each search index.

## Additional Information

- How to Install a Sphinx Engine

## Connect with Sphinx Engine installed on Windows server

For make extension work with Sphinx Engine installed on **Windows based server**, follow these steps: Before you generate a Sphinx configuration file, you must ensure that you have correct settings for Sphinx Engine connection to the database. Go to [magento\_root]/app/etc/ and edit **local.xml** file.

In the line <host> set IP address of the database server: <default\_setup> <connection>

<host>85.75.44.153</host> ~! Don't use **localhost** or **127.0.0.1** as server IP address.

You need to change host IP address even if your Magento and database are placed on the same server.

1. Go to **System > Configuration > Mirasvit Extensions > Sphinx Search**.
2. Fill in following fields:
  - **Base Path** - any folder on sphinx server, where you would like to store sphinx configuration and indexes (ex. /var/sphinx/)
  - **Sphinx Host** - sphinx daemon host
  - **Sphinx Port** - sphinx daemon port
3. Press button **Save Config**.
4. Press button **Generate Config file**.
5. Copy generated config file to the **Base Path** on server.
6. Run following commands:
  - indexer --config <base\_path>/sphinx.conf --rotate mage\_catalog\_product. Where mage\_catalog\_product code of index. If you use few indexes (products, categories, cms pages etc), you need do this command for each index.
  - searchd --config <base\_path>/sphinx.conf
7. If configuration is correct you will see search queries in sphinx log file (<base\_path>/searchd\_query.log)
8. After done key configuration, you should configure cronjob for automatically update sphinx indexes.
  - Full reindex indexer --config <base\_path>/sphinx.conf --rotate mage\_catalog\_product
  - Delta reindex indexer --config <base\_path>/sphinx.conf --rotate delta\_mage\_catalog\_product indexer --config <base\_path>/sphinx.conf --merge mage\_catalog\_product delta\_mage\_catalog\_product --merge-dst-range deleted 0 0 --rotate
  - Start sphinx daemon searchd --config <base\_path>/sphinx.conf
  - Stop sphinx daemon killall -9 searchd
  - Restart sphinx daemon killall -9 searchd; searchd --config <base\_path>/sphinx.conf;

**Full example of crontab** 30 \* \* \* \* indexer --config sphinx.conf --rotate delta\_mage\_catalog\_product; indexer --config sphinx.conf --merge mage\_catalog\_product delta\_mage\_catalog\_product --merge-dst-range deleted 0 0 --rotate; killall -9 searchd; searchd --config sphinx.conf; 0 2 \* \* \* indexer --config sphinx.conf --rotate mage\_catalog\_product; killall -9 searchd; searchd --config sphinx.conf; First job run delta reindex each hour (00:30, 01:30, 02:30 etc) and restart sphinx after delta reindex. Second job run full reindex every day at 02:00 and

restart sphinx daemon. If you have few search indexes, you should run indexer command for each search index.

## Additional Information

- How to Install a Sphinx Engine

# "Long Tail" search (configure a search by SKU, abbreviations etc)

To receive correct search results for words that contain dashes or any other non-alphabetic symbols, we recommend to use **long-tail** search feature. Go to **System > Configuration > Mirasvit Extensions > Sphinx Search**. In the section **Merge Settings (Long tail)** setup regular expressions to receive required search results.

- **Match Expression** - the regular expression(s) that parses words for the further replacing. Parsing goes for search index, during an indexing process, and goes for search phrases during search. E.g. `/([a-zA-Z0-9]*[\-\/][a-zA-Z0-9]*[\-\/]*[a-zA-Z0-9]*)/`
- **Replace Expression** - the regular expression(s) to parse characters to be replaced. Parsing goes in results of "Match Expression". E.g. `/[\-\/]/`
- **Replace Char** - the character to replace values founded by "Replace Expression". E.g. empty value.

**How it works?** During reindex extension match search index by **Match Expression** and then replace chars in matches by **Replace Expression** to **Replace Char**. Same actions extension do for each user search phrase. As result, search index and search query contains same search phrase form. For example, we have a product "Canon PowerShot SX500 IS". If customer is searching for the "Canon PowerShot SX-500IS", default search will not find the mentioned product. Using "long-tail" search, a search function recognizes a characters combination by pattern during reindex and search (by **match expression**) and replaces special characters (**replace expression**) to the empty ones or any other characters (**replace char**). So, during reindexing the "SX500 IS" will be converted to the "SX500IS" and during the search the "SX-500IS" will be converted to the "SX500IS" aswell. As a result the search will be able to find product by several combinations of product's name spelling. **Example 1**

- **Match Expression** - `/[a-zA-Z0-9]*-[a-zA-Z0-9]*/` Matched text: SX500-123, GLX-11A, GLZX-VXV, GLZ/123, GLZV 123, CNC-PWR1
- **Replace Expression** - `/-/`
- **Replace Char** - empty

Result text: SX500123, GLX11A, GLZXVXV, GLZ/123, GLZV-123-123, CNCPWR1 **Example 2**

- **Match Expression** - `/[a-zA-Z0-9]*[\-\/][a-zA-Z0-9]*/` Matched text: SX500-123, GLX-11A, GLZX-VXV, GLZ/123, GLZV 123, CNC-PWR1
- **Replace Expression** - `/[\-\/]/`
- **Replace Char** - empty

Result text: SX500123, GLX11A, GLZXVXV, GLZ123, GLZV123, CNCPWR1 **Example 3**

- **Match Expression** - `/[a-zA-Z0-9]*[-\s][a-zA-Z0-9]*([-\s][a-zA-Z0-9]*)?/` Matched text: SX500-123, GLX-11A, GLZX-VXV, GLZ/123, GLZV-123-123, CNC-PWR1
- **Replace Expression** - `/[-\s]/`
- **Replace Char** - empty

Result text: SX500123, GLX11A, GLZXVXV, GLZ123, GLZV123123, CNCPWR1

## Manage Stopwords

Stopwords are words which have little lexical meaning or ambiguous meaning and are not useful for search (ex. and, or, the, a, with, etc). These words will be removed from search phrase. To manage stopwords go to **Search > Dictionary of Stopwords**. You can add new stopwords manually or can import them from CSV file. To import stopwords press button **Import Dictionary**. You'll see a dropdown list with available dictionaries. Select dictionary, store view and press button **Import Dictionary**. All dictionaries are located in the folder `[magento_root]/var/sphinx/stopwords/`. By default extension has dictionary of English stopwords (665 items). You can create a custom dictionary for your own language using the same file format. Also stopwords dictionaries for different languages can be found in web.

### Dictionaries

- English
- French
- Dutch
- German
- Polish

### Useful links

- <http://en.wikipedia.org/wiki/Stopword>

## Manage Synonyms

Synonyms are words with the same or similar meanings. To manage synonyms go to **Search > Dictionary of Synonyms**. You can add new synonyms manually or can import them from CSV file.

### Import Synonyms

## Import

Dictionary \*

English.txt ▼

Store View \*

Main Website

Main Store

English

French

German

nikastic.com

Second Website

Second Store

Second Store View

To import synonyms, follow these steps:

1. Go to **Search > Dictionary of Synonyms**
2. Press button **Import Dictionary**.
3. You'll see a dropdown list with available dictionaries. Select dictionary, store view and press button **Import Dictionary**.

All dictionaries are located in the folder [magento\_root]/var/sphinx/synonyms/. By default extension has dictionary of English synonyms (60370 items). You can create a custom dictionary for your own language using the same file format. Also synonyms dictionaries for different languages can be found in web.

## Dictionaries

- English
- Czech
- Spanish
- French
- Portuguese
- Dutch

## Add new synonyms manually

## General Information

Synonyms \*

zymolysis,zymosis,fermentation,fermenting,fermen  
t

Store View \*

English

To add new synonyms, follow these steps:

1. Go to **Search > Dictionary of Synonyms**
2. Press button **Add New**.
3. You'll fill synonyms textarea (**coma as separator**) and select store view.
  - o Number of synonyms must be greater than 1
  - o Synonym can contain only one word
  - o The lenght of synonym must be greater than 1 char
4. Press **Save**

Synonym (synonyms) are single words, i.e. without spaces, dashes, slashes etc. In other cases, you need use **Replace words in search query** (see Global Search Options).

## Change the possition of the product or the category in search results

Usually if you need to improve relevance of the search, you should change a global search settings. But in some cases you need to move up/down only individual product or category in search results. To do this, open product or category edit page and find attribute **Search Weight**. By changing this attribute, you can change the possition of individual product or category in search results. You can set it's value from **-100** (product or category always will be in the bottom of search results list) to **100** (product or category always will be in the top of search results list). Also is available ability to set additional weight for any individual product or category (products in this category). Extension addeds new attribute **Search Weight** at product edit page and category edit page. Additional priority must be from -100 (always last in the search results) to 100 (always first in the search results).



# Manage search landing pages

Landing search page is the page with predefined URL that returns search results by predefined search phrase. Using this feature, you can create a page `http://store.com/black-samsung-phone.html` and link it with search phrase **black samsung phone**. When customer visits this page, he will see search results by phrase **black samsung phone**. Also you can set up Meta tags and H1 for this page. Go to **Search > Manage Landing Pages**. Here you can manage landing pages. Press the button **Add new** and fill in following fields:

Edit Landing Page 'All Phones'

Back

Reset

Delete

Save

Save And Continue Edit

## General Information

Search Phase \*

phone

Landing Url \*

all-phones

▲ ex. phones/samsung-phone

Active \*

Yes

Title \*

All Phones

Meta Title

All Phones

Meta Keywords

phone, cell phone

Meta Description

meta description

Layout

```
<reference name="root">
 <action method="setTemplate"><template>page/1column.phtml</template></action>
</reference>
```

- **Search Phrase** - the phrase for search (ex. samsung phone)
- **Landing URL** - the URL of landing page without first slash (ex. phones/samsung.html)
- **Title** - the title (H1) of the page
- **Meta title**
- **Meta Keywords**
- **Meta Description**
- **Layout** - xml layout (update section)

# Configure search indexes

Go to **Search > Manage Search Indexes**. You can add new index, edit indexes, run reindexing process.

## Manage Search Indexes

[+ Add New](#)Page [1](#) of 1 pages | View [20](#) per page | Total 7 records found[Reset Filter](#)[Search](#)

Select All

|

Unselect All

|

Select Visible

|

Unselect Visible

|

0

Actions

Submit

items selected

	Title	State	Action
Any			
<input type="checkbox"/>	Products	READY	<a href="#">Run Reindex</a>
<input type="checkbox"/>	Categories	READY	<a href="#">Run Reindex</a>
<input type="checkbox"/>	Our Blog	DISABLED	<a href="#">Run Reindex</a>
<input type="checkbox"/>	Search Terms	READY	<a href="#">Run Reindex</a>
<input type="checkbox"/>	Pages	READY	<a href="#">Run Reindex</a>
<input type="checkbox"/>	Forum	REINDEX REQUIRED	<a href="#">Run Reindex</a>
<input type="checkbox"/>	Our Offers	READY	<a href="#">Run Reindex</a>

After creating or updating any index you must reindex this index. By default, extension supports search by these indexes:

- Magento
  - Catalog Products
  - Catalog Categories
  - Catalog Attribute
  - Cms Pages
  - Catalog Search Queries

Also extension supports external indexes:

- Blogs and Forums
  - Wordpress Blog
  - VBulletin Forum
  - Simple Press Forum
  - News by CommerceLab
- Mirasvit
  - Product Questions
  - Knowledge Base
- AheadWorks
  - Blog
  - Knowledge Base
- MageBuzz
  - Mageticket / Knowledge base
- Magpleasure
  - Blog Posts
- Templates Master
  - Helpmate
- FMEEExtensions

- Fags + Product Ask a Questions
  - Shop By Manufacturers/Brands
- Magentothem
  - Blog Posts
- Additionally you can add own search index

External indexes will work only if you install related to these indexes software.

## Configure a search by products

Go to **Search > Manage Search Indexes**, open **Product index**. Select store view for which **Product index** will be applied.

- **Title** - title of the search index. Also this title will be displayed in the search result tab for the selected store.
- **Attributes** - for each searchable attribute you can set own weight. Different weights indicate the importance of each attribute for product relevancy. The maximum weight is 100 (highest priority), the minimum weight is 0.
- **Search by parent categories names** - if option enabled, customer can find a product by it's parent categories
- **Search by child products attributes (for bundled and configurable products)** - if option enabled, customer can find bundled or grouped product by information from it's associated products.
- **Seach by product tags** - if option enabled, customer can find product by related tags.
- **Seach by product id** - if option enabled, customer can find product by product id (entity\_id).
- **Search by custom options** - if option enabled, customer can find product by custom options (SKU and title) (from Mar 21, 2014).
- **Push "out of stock" products to the end** - if option enabled, all out of stock products will be displayed at bottom of product list. Extension performs search by **all searchable attributes** (with and without weights). For mark/unmark attribute as searchable, go to **Catalog > Attributes > Manage Attributes**, find certain attribute, change option **Use in Quick Search** and press **Save Attribute**. After changing searchable attributes, you need run search reindex at **System / Index Management**.

Choose Store View:
All Store Views

Edit Search Index 'Products'
Back
Save
Save And Continue Edit

### General Information

Index \*
Catalog Products

Title \*
Products

### Attributes

Attribute	Weight	
Name	100	
SKU	80	
Manufacturer	50	
Model	30	
Category	10	
Add attribute		

### Additional Search Index Configuration

Search by parent categories names \*
Yes

Search by child products attributes (for bundle and configurable products) \*
No

Search by product tags \*
Yes

Search by product id \*
Yes

Search by custom options
Yes

Push "out of stock" products to the end
Yes

## Configure a search by categories

Go to **Search > Manage Search Indexes** and press the button **Add New**. From the dropdown list select **Catalog Categories**, fill in fields.

- **Title** - title of search index.
- **Position** - sets the position of the index in the search results.
- **Attributes** - for each searchable attribute you can set own weight. Different weights indicate the importance of each attribute for product relevancy. The maximum weight is 100 (highest priority), the minimum weight is 0.

## General Information

Index *	Catalog Categories
Title *	Categories
Position *	10
Active *	Yes

## Attributes

Attribute	Weight	
Name	100	
Meta Title	80	
Meta Keywords	1	
Meta Description	1	
Description	1	
		Add attribute

Extension performs search by **all searchable attributes** (with and without weights). For mark/unmark attribute as searchable, go to **Catalog > Attributes > Manage Attributes**, find certain attribute, change option **Use in Quick Search** and press **Save Attribute**. After changing searchable attributes, you need run search reindex at **System / Index Management**. After the index is created, add the category attributes to the list of searchable and set weights for them. The attribute's weight indicates the importance of the attribute for category relevancy. The maximum weight is 100 (highest priority), the minimum weight is 0.

## Configure a search by attributes

Extension allows you to get search results by any attribute. Go to **Search > Manage Search Indexes** and press the button **Add New**. Select from the dropdown list **Catalog Attribute**, fill in the following fields:

- **Title** - Title of the search index.
- **Position** - Sets the position of the index in the search results.
- **Attributes** - Attributes, which uses to set weight for search. There is only **Option Label** attribute available.
- **Attribute** - Select attribute you want to include in search results.
- **Url Template** - URL for your attribute search results with dynamical variables.

Extension allows to create only one search attribute. **Example**

Select attribute from **Attribute** list and fill in **Url Template** with store search URL including attribute

code. Use attribute variable for the **Url Template: Attribute:** Brand [computer\_manufacturers]  
**Url Template:** http://example.com/catalogsearch/advanced/result/?computer\_manufacturers={value} ~!  
Search results by attribute occurs only via **advanced search**. Therefore, Url of the template must contain line **/advanced/**.

**Edit Search Index 'Brand'**

General Information

Index \*

Catalog Attribute

?

Title \*

Brand

?

Position \*

30

?

Active \*

Yes

Attribute

Attribute \*

Url Template \*

Attributes

Attribute	Weight	
Option Label (label)	1	<div></div>
<div><div></div>Add attribute</div>		

Search

[Home](#) / [Catalog Advanced Search](#) / **Results**

## Catalog Advanced Search



2 Item(s) were found using the following search criteria

→ **Brand:** Sony

Don't see what you're looking for? [Modify your search](#)

2 Item(s) Show 9 per page

View as: **Grid** [List](#) Sort By Name ↑

 <p><u><a href="#">Sony VAIO 11.1" Notebook PC VGN-TXN27N/XARX</a></u></p>	 <p><u><a href="#">Sony VAIO VGN-TXN27N/B 11.1" Notebook PC</a></u></p>		
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

## Create custom search index

To perform search by custom entities content, follow these steps:

1. Go to `app/code/local/Mirasvit/SearchIndex/Model/Index/`
2. Create folder/file structure
  - `<Company>/<Extension>/<EntityType>/Index.php`
  - `<Company>/<Extension>/<EntityType>/Indexer.php`
 i.e. `app/code/local/Mirasvit/SearchIndex/Model/Index/Mirasvit/News/News/Index.php` or `app/code/local/Mirasvit/SearchIndex/Model/Index/Aw/Kb/Article/Indexer.php`
3. Copy content from `../Mirasvit/Action/Action/Index.php` to your index file
  1. Rename class
  2. Change result of method `getBaseGroup` to `<Company>`
  3. Change result of method `getBaseTitle` to `<Extension>`
  4. Set primary key in method `getPrimaryKey`
  5. Change method `canUse` (it must return TRUE)

6. Set searchable attributes in method `getAvailableAttributes`
7. Set correct interface to fetch collection of items in method `getCollection`
4. Copy content from `../Mirasvit/Action/Action/Indexer.php` to your indexer file
  1. Rename class
  2. Set correct interface to fetch collection of items in method `_getSearchableEntities`
5. If all correct, you can add index at **Search > Manage Search Indexes**
6. Add changed to layout file `searchindex.xml`
7. Create new template at  
`<package>/<theme>/template/searchindex/index/<company>/<extension>/<entitytype>.phtml`

## Create a connection to another database

To create a new connection add a new connection code to **app/etc/local.xml** after primary database section. Check the following example of **local.xml**:

```
<?xml version="1.0"?> <config> <global> <install>
<date><![CDATA[]]></date> </install> <crypt> <key><![CDATA[]]></key> </crypt>
<disable_local_modules>false</disable_local_modules> <resources> <db>
<table_prefix><![CDATA[]]></table_prefix> </db> <default_setup> <connection>
<host><![CDATA[host]]></host> <username><![CDATA[username]]></username>
<password><![CDATA[password]]></password> <dbname><![CDATA[dbname]]></dbname>
<initStatements><![CDATA[SET NAMES utf8]]></initStatements>
<model><![CDATA[mysql4]]></model> <type><![CDATA[pdo_mysql]]></type>
<pdoType><![CDATA[]]></pdoType> <active>1</active> </connection> </default_setup> <!--
START new connection to wordpress database !--> <wordpress_setup> <connection>
<host><![CDATA[wordpress_host]]></host>
<username><![CDATA[wordpress_database_username]]></username>
<password><![CDATA[wordpress_database_password]]></password>
<dbname><![CDATA[wordpress_database_name]]></dbname> <initStatements><![CDATA[SET
NAMES utf8]]></initStatements> <model><![CDATA[mysql4]]></model>
<type><![CDATA[pdo_mysql]]></type> <pdoType><![CDATA[]]></pdoType> <active>1</active>
<use>0</use> </connection> </wordpress_setup> <!-- FINISH new connection to wordpress database !--
> </resources> <session_save><![CDATA[files]]></session_save> </global> <admin> <routes>
<adminhtml> <args> <frontName><![CDATA[admin]]></frontName> </args> </adminhtml>
</routes> </admin> </config>
```

After changing the `local.xml` file, you must clear the magento cache.

## Configure a search by Wordpress blog posts

Go to **Search > Manage Search Indexes** and press the button **Add New**. From the dropdown list select **Wordpress Blog** and fill in fields. Press **Save and Continue Edit**.



## Edit Search Index 'Our Blog'

[Back](#) [Delete](#) [Save](#) [Save And Continue Edit](#)

General Information	Database Settings
<p>Index * <input type="text" value="Wordpress Blog"/></p> <p>Title * <input type="text" value="Our Blog"/></p> <p>Position * <input type="text" value="20"/></p> <p>Active * <input type="text" value="Yes"/></p>	<p>Database Connection Name * <input type="text" value="wordpress_setup"/></p> <p>Table Prefix <input type="text" value="wp_"/></p>
	URL Settings
	<p>Url Template * <input type="text" value="/blog/{slug}.html"/></p> <p>▲ /blog/?p={id}</p>

Attributes		
Attribute	Weight	
<input type="text" value="Post Title"/>	<input type="text" value="100"/>	<a href="#">✕</a>
<input type="text" value="Post Content"/>	<input type="text" value="20"/>	<a href="#">✕</a>
<input type="text" value="Post Excerpt"/>	<input type="text" value="50"/>	<a href="#">✕</a>
<a href="#">+ Add attribute</a>		

After the index is created, you can change it's settings:

- **Title** - the title of the search index (displayed in the tab).
- **Position** - the position of the tab to display in search results.
- **Active** - if it is active, the search index will be displayed in search results.
- **Attributes** - the post's attributes with their weights be used for search.
- **Table Prefix** - the prefix for wordpress tables. if in database table names are like **wp\_posts**, then prefix should be **wp\_**.
- **Base Url** - the full URL for your posts with dynamical variables:
  - {ID}
  - {post\_author}
  - {post\_date}
  - {post\_date\_gmt}
  - {post\_title}
  - {post\_status}
  - {post\_type}
  - {post\_name}
  - {guid}
  - {category\_name}
  - {category\_slug}
  - {Y} - post date year, e.g. 2018
  - {m} - post date month, e.g. 08
  - {d} - post date day, e.g. 23

Typical base urls:

- [http://example.com/blog/{post\\_name}.html](http://example.com/blog/{post_name}.html)

- `http://example.com/blog/?p={ID}`
- `http://example.com/{category_slug}/{post_name}.html`
- **Database Connection Name** - the connection name for wordpress database. If wordpress is installed on same database, the correct value is **default\_setup**. If wordpress uses another database, then you need to create a new connection in **app/etc/local.xml** and to set a database connection name to the name of this connection. To create a new connection see [Create a connection to another database](#).

## Configure Search Results

Extension can work using one of following engines:

- **Built-in search engine** - will be used internal search algorithm of our extension. This mode does not require installation of Sphinx Engine on your server. You'll receive the same features as with Sphinx Engine. The only drawback is a slower speed of the search than with the Sphinx Engine.
- **External search engine** - will be used Sphinx Engine installed on the server of your store. See also [Connect with Sphinx Engine](#).
- **External search engine (another server)** - will be used Sphinx Engine installed on any other server.

- If you use **External search engine** or **External search engine (another server)** mode, **Attributes** of the Search indexes are not applied in the search results relevancy. -

## Quick Start Configuration

1. Go to **System > Configuration > Mirasvit Extensions > Sphinx Search**. Select Search Engine. Press button **Save Config**.

## Sphinx Search

Configuration

Search Engine

Built-in search engine

?

[GLOBAL]

Advanced Settings

Max number of items in the result

0

?

[GLOBAL]

Match Mode

Matches query as an expression

?

[GLOBAL]

Search Template

AND matching (word1 & word2 & word3...)

?

[GLOBAL]

Enable wildcard search

Disable

?

[GLOBAL]

Wildcard Exceptions

+ Add

?

[GLOBAL]

Replace words in search query

+ Add

?

[GLOBAL]

"Not" words

+ Add

?

[GLOBAL]

Merge Settings (Long tail)

For example, we have a product "Canon PowerShot SX500 IS". If customer searches for 'Canon PowerShot SX-500IS', default search will find "Canon PowerShot SX-500IS" but not "Canon PowerShot SX500 IS". Using "long-tail" search, the search function recognizes char combination pattern during reindex and search (by **match expression**). So during reindexing "SX500 IS" will be converted to "SX500IS" and during search "SX-500IS" will be converted to "SX500IS".

Match Expression

/[a-zA-z0-9]\*-[a-zA-z0-9]\*/

?

[GLOBAL]

Replace Expression

/-/

?

[GLOBAL]

Replace Char

?

[GLOBAL]

- If you use **External search engine**, press button **Run Full Reindex**, then press button **Start Sphinx daemon**. See also Connect with Sphinx Engine. If you use **External search engine (another server)**, fill in fields, press button **Generate Config file** and copy config file to your Base Path. See also Connect with External Sphinx Engine.
- Go to **System > Index Management**. Run reindex of the **Search Index**.

## Advanced Configuration

To receive the best search results, you should configure global search options and search options specifically for each search index type (products, categories, blog posts). Check following links for detailed information:

- Configure a global search options
- Configure search indexes

# How to run search reindex via command line interface

## How to reindex Magento search indexes

To reindex search indexes (and indexes of Sphinx Engine, if you use it) via command line (shell, SSH) go to the magento root folder and use this command: `php shell/indexer.php --reindex catalogsearch_fulltext`

## How to reindex search indexes for Sphinx Engine installed on the same Magento server.

Go to <magento\_root>/shell folder and use php script search.php: ~! Commands don't work with "Built-in Search Engine" mode. Usage: `php search.php [options] --engine` Get Current Engine Class Name --reindex Run sphinx reindex --reindex-delta Run sphinx delta reindex Examples `php search.php --engine php search.php --reindex-delta`

## How to import synonyms and stopwords

Go to <magento\_root>/shell folder and use php script search.php: Usage: `php search.php [options] --import-synonyms <csv file>` Import synonyms --import-stopwords <csv file> Import synonyms Examples `php search.php --import-synonyms /dir/dir/en.txt php search.php --import-stopwords /dir/stopwords_nl.txt`

## Why speed of search is too slow?

Speed of search is directly related to the store speed and can't be faster than speed of the store product page opening.

To check the speed of search by extension, go to **Search > Validate Search Results** and click on the tab **Validate Speed**. Fill in the **Search Phrase** field with the search phrase and click button **Submit**.

You will get the **Search Time (seconds)** of the search query and **Number of Results**.

Validate Result

Validate Speed

## Search Validation

### Check why product can't be found by search phrase

Search Phrase \*

Store



Search Time (seconds)

0.0037

Number of Results

6

Current Search Engine

 You use "Built-in Sphinx Search Engine".

#### Product ID

16

166

20

19

18

17

If **Search Time** is much faster then current search speed on the store, **Current Search Engine** works correctly. It means that slow speed of search happens due to low speed of the loading store page: a lot of time takes loading of resources. Therefore, you can increase the speed of search only via more precise configuration of the store.

## The ways to increase speed of search:

- 1) If you have a large product catalog (more than 10 000 products), we suggest to use **Search Sphinx Engine**. See how to Install a **Sphinx Engine**
- 2) Set **Max number of items in the result**
- 3) Solve all issues which appear at the validator page. Path of the validator page: Configuration > System > Mirasvit Extensions > System > Installed Extensions. Click on the button Run validation tests for all extensions.
- 4) Try to set the **"true"** value at the variable `$_useNativeTables` in the file located at `Magento_root/app/code/local/Mirasvit/SearchIndex/Model/Index.php` on the line 21.

## Factors on which speed of load depends on

- 1) Store cache configuration
- 2) Number of products in catalog

- 3) Conflicts with custom extensions in Magento core
  - 4) Power of your server
  - 5) Usage of shared/managed server on which store is located
  - 6) The traffic flow
  - 7) The server location
  - 8) Type and speed of the connection
- etc.

## Troubleshooting

- **I have Magento Enterprise Edition, and search does not return correct results** This issue can appear, when you're using simultaneously several search modules, and they're conflicting with each other. **Solution:**
  1. First of all, make sure, that **there's no active third-party search extensions**. You can check it at `/app/etc/modules` folder, where each module has it's respective XML file.
  2. If there any, except of **Mirasvit** or standard **Mage** extensions, turn them off either by renaming their respective file at `/app/etc/modules`, or editing them.
  3. Turn off extensions **Enterprise\_Search** and **Enterprise\_CatalogSearch**, if they're enabled.
  4. Purge ALL cache, that you're using, including external.
  5. Run reindex.
- **Configurable Swatches are not shown for the products on the search result pages** Error could appear if module Configurable Swatches is enabled.  
 This problem occurs because Magento module is loaded before our search module. **Solution:**  
 To solve this problem you should tell Magento load the module **Mage\_ConfigurableSwatches** after our module **Mirasvit\_SearchIndex**  
 For this open the file `app/etc/modules/Mirasvit_SearchIndex.xml` and add dependency for module `Mage_ConfigurableSwatches`.  
 After changes the file `app/etc/modules/Mirasvit_SearchIndex.xml` should look as follows:
 

```
<config> <modules> <Mirasvit_SearchIndex> <active>true</active>
<codePool>local</codePool> </Mirasvit_SearchIndex> <Mage_ConfigurableSwatches>
<depends> <Mirasvit_SearchIndex/> </depends> </Mage_ConfigurableSwatches> </modules>
</config>
```
- **Mana Filters do not work on the search result page after installing our extension** There is a known conflict with the extension **Mana\_Fitlers**. Error may occur on the search result page, when trying to apply filters. **Solution:**

1. **app/design/frontend/[YOUR PACKAGE]/[YOUR THEME]/layout/catalogsearch.xml**- find and comment out everything inside <reference name="content"> under <catalogsearch\_result\_index> as shown below: <catalogsearch\_result\_index> ... <reference name="content"> <!-- COMMENT EVERYTHING IN IT --> </reference> ... </catalogsearch\_result\_index>
  2. **app/etc/modules/Mirasvit\_SearchIndex.xml** – make sure that this module loads before our module by adding dependency on the **Mana\_Filters** module: <config> <modules> <Mirasvit\_SearchIndex> <active>true</active> <codePool>local</codePool> </Mirasvit\_SearchIndex> <Mana\_Filters> <depends> <Mirasvit\_SearchIndex/> </depends> </Mana\_Filters> </modules> </config>
- **Conflict with Netzarbeiter\_GroupsCatalog2** This conflict occurs because both extensions rewrites the same core Magento model class.  
To solve the conflict you should change the file **app/code/community/Netzarbeiter/GroupsCatalog2/Model/CatalogSearch/Resource/Fulltext/Collection.php**, specifically change the string extends Mage\_CatalogSearch\_Model\_Resource\_Fulltext\_Collection to make it look as follows: extends Mirasvit\_SearchIndex\_Model\_Catalogsearch\_Resource\_Fulltext\_Collection.  
That's it, the search should work properly now.
  - **Conflict with Simtech\_Searchanise** This conflict occurs because both extensions rewrites the same core Magento model class.  
To solve the conflict you should change the file **app/code/community/Simtech/Searchanise/Model/Resource/Fulltext/Collection.php**, specifically change the string extends Mage\_CatalogSearch\_Model\_Resource\_Fulltext\_Collection to make it look as follows: extends Mirasvit\_SearchIndex\_Model\_Catalogsearch\_Resource\_Fulltext\_Collection.  
That's it, the search should work properly now.
  - **Conflict with MageWorx\_CustomOptions** This conflict occurs because **MageWorx\_CustomOptions** extensions set additional parameters to MySql query. **Solution:** **/app/code/community/MageWorx/CustomOptions/Model/Observer.php** - find the following methods:
    - public function catalogProductCollectionLoadBefore
    - public function catalogProductCollectionLoadAfter
 and place the following code at the beginning of method : if (Mage::app()->getRequest()->getModuleName() == 'catalogsearch' || Mage::app()->getRequest()->getModuleName() == 'searchautocomplete') { return; } to make it look as follows : public function catalogProductCollectionLoadBefore(\$observer) { if (Mage::app()->getRequest()->getModuleName() == 'catalogsearch' || Mage::app()->getRequest()->getModuleName() == 'searchautocomplete') { return; } ... } public function catalogProductCollectionLoadAfter(\$observer) { if (Mage::app()->getRequest()->getModuleName() == 'catalogsearch' || Mage::app()->getRequest()->getModuleName() == 'searchautocomplete') { return; } ... }
  - **Thread stack overrun MySql error** This error occurs because you have too much search attributes or your thread\_stack size is not enough to process query.  
**Solution:**
    1. You can decrease search attributes qty in Search-> Manage Search Indexes;
    2. You can increase your thread\_stack size. Minimum suggested value is 256KB. You can add

the `--thread_stack` option with the correct value to the command-line when starting the server, or you could add it to the configuration file (`my.cnf` or `my.ini`, depending on your system).

## Change Log

# Change Log

## 2.3.47

(2018-12-06) Fixed

- improved conflicts validator [#124]

## 2.3.46

(2018-12-03) Fixed

- made synonyms store-specific [#80]
- search by tags indexing issue

## 2.3.45

(2018-11-08) Fixed

- Synonyms results have search weight = 0

## 2.3.44

(2018-11-06) Fixed

- 404 to search notice issue

## 2.3.43

(2018-10-30) Fixed

- `getSize` issue after group by

## 2.3.42

(2018-10-25) Fixed

- Issue with tags indexing

## 2.3.41

(2018-10-17) Fixed



- Issue with search by custom options

## **2.3.40**

(2018-10-17) Features

- Added "Minimal Query Length" option to Validate Search Results

## **2.3.39**

(2018-10-16) Fixed

- getSize() issue after group by
- wrong link in validate search results

## **2.3.38**

(2018-09-27) Fixed

- Compatibility with PHP7.2

Features

- Added search autocomplete profiler

Documentation

- Info about date modifiers for Wordpress URL template

## **2.3.37**

(2018-08-14) Improvements

- Ability to use date modifiers to form Wordpress blog post URL

Fixed

- changed error message label to info

## **2.3.36**

(2018-07-25) Fixed

- fixed generating sphinx config file issue

Documentation

- Replace words in search query

## **2.3.35**

(2018-07-20) Improvements

- added sphinx3.\* support
- improved installation validator functionality (better check for conflicts)

Fixed

- changed status caption in validator search results
- fixed negative search weights support for sphinx engine mirasvit/extension\_searchsphinx#53

## 2.3.34

(2018-06-14) Fixed

- Issue with search by child products attributes

## 2.3.33

(2018-06-14) Fixed

- search by product ID issue

## 2.3.32

(2018-03-09) Fixed

- Remove all non alphanum chars from sphinx bin path

## 2.3.31

(2018-02-07) Improvements

- Addition validator message regarding product visibility in search results
- Improved search algorithm for External Mode (Sphinx Engine)

Fixed

- Fix error 'headers already sent'

## 2.3.30

(2018-01-11) Improvements

- Show validation message for conflict with the module 'Simtech\_Searchanise' mirasvit/extension\_searchultimate#5

## 2.3.29

(2018-01-10) Improvements

- Improve validator message

### **2.3.28**

(2017-11-07) Fixed

- Properly index searchable data

### **2.3.27**

(2017-10-30) Feature

- Add search terms export

Improvements

- Make external mode more accurate

### **2.3.26**

(2017-09-04) Fixed

- Inaccurate weight to items that do not match the search word in external mode

Improvements

- Added stopwords dictionaries for German, French, Dutch

### **2.3.25**

(2017-04-26) Fixed

- Fix error 'Headers already sent' caused by function '404 to search'

### **2.3.24**

(2017-04-21) Fixed

- Fix module Mage Configurable Swatches does not work on search result pages when our extension used (affects all).

Improvements

- Troubleshoot. Fix problem with the Mage Configurable Swatches module.
- Troubleshoot. Fix conflict with Mana Filters extension.

### **2.3.23**

(2017-02-15) Fixed

- Escape sphinx query

Improvements

- Improve performance for single result redirect

### **2.3.22**

(2017-01-24) Fixed

- Fix singularize issue in Dutch language (affects all)

### **2.3.21**

(2016-11-02) Fixed

- Compatibility with Magento 1.9.3.1
- Use product search weight from default scope if it empty at a store scope

### **2.3.20**

(2016-11-02) Documentation

- Documentation updates

### **2.3.19**

(2016-11-02) Fixed

- Wrong search weight calculation for products

### **2.3.18**

(2016-10-31) Feature

- Ability to select ignored categories for the search index

Fixed

- Wildcard exceptions ignored if wildcard suffix or prefix mode are used (affects all)

### **2.3.17**

(2016-10-13) Fixed

- Correctly use translation method

### **2.3.16**

(2016-10-11) Fixed

- Fix problem with search terms when flat catalog product enabled

Improvements

- Performance improvement for single result redirect

### **2.3.15**

(2016-08-18) Improvements

- Improve search speed script

Fixed

- Fix CRC issue for search index files

## **2.3.14**

(2016-08-10) Fixed

- Fix for php 7 constructor method name same as class name

Improvements

- Surface improvements for the indexation process

## **2.3.13**

(2016-08-04) Improvements

- Scope label for settings at the search indexes
- Improve search for index Mirasvit Gift Registry
- Change column type for search index field "title"

## **2.3.12**

(2016-08-01) Improvements

- Show debug info for External Sphinx Engine
- Stop searchd using its own command and use URL of default store

Improvements

- Validate extension's CRC values

## **2.3.11**

(2016-07-13) Fixed

- Fixed an issue with mb\_strtolower

## **2.3.10**

(2016-07-05) Improvements

- NL translation

## **2.3.9**

(2016-07-05) Fixed

- Check Giftr block existence
- Fixed a potential XSS issue with m\_test\_search\_speed.php

## **2.3.8**

(2016-06-24) Improvements

- New search index for Mirasvit Gift Registry

## **2.3.7**

(2016-06-21) Fixed

- Properly process multibyte UTF-8 characters

### **2.3.6**

(2016-06-13) Fixed

- Fix issue with the search queries that contain the word 'as'

### **2.3.5**

(2016-05-31) Improvements

- Enable morphology search for External Sphinx Engine when the wildcard search is enabled

### **2.3.4**

(2016-05-28) Fixed

- Convert store ID to array only for Sphinx Client
- Rebuild "Search terms" section and improve performance

### **2.3.3**

(2016-05-23) Improvements

- Added ability to use custom attributes in search by categories

### **2.3.2**

(2016-05-13) Features

- Added ability to filter search results by relevance

Improvements

- Improved query builder performance (stopwords)
- New search index for Blackbird ContentManager
- Compatibility with the version of Magento 1.4.1.1
- Added bundled product to primary product search index
- Added store code to multi-store search results configuration
- Extended list of conflicted extensions for validator (Activo/CatalogSearch, Netzarbeiter/GroupsCatalog2)

Fixed

- Fixed an issue with duplicating queries at Catalog / Search Terms
- Fixed an issue with enclose relevance column
- Fix problem with search report (affects from 2.3.2.100)
- Fixed issue with "." in search terms (sphinx mode)
- Update composer.json (requirements)
- Fixed an issue with Undefined offset for entity custom weights
- Fixed an issue with wrong quote in sql query, when option "Push out of stock products to bottom" is enabled