

# Search Autocomplete and Suggest

- Installation
  - Upgrading from version 1.x
- How does it work?
- Configuration
  - Basic representation settings
  - Search item template
  - Search query configuration
- Uninstallation
- Troubleshooting

**i** Find the answer to your question!  
Check out our [Knowledge Base](#).

**i** Looking for the same Magento 2 functionality?  
Check out our [Advanced Search for Magento 2](#):

- [Product page](#)
- [User Guide](#)

Extension page: <http://ecommerce.aheadworks.com/magento-extensions/search-autocomplete-and-suggest.html>

Improve your store usability with the Search Autocomplete and Suggest extension by aheadWorks. This software makes it simple for your customers to find exactly what they need quickly and conveniently.

**!** For Advanced Search users

- If you are using the 1.4.0 version of the [Advanced Search](#) extension, make sure that you update Search Autocomplete And Suggest to version 3.3.5+

## Installation

1. Backup your web directory and store database.

▼ [Click to view details](#)

You can make backup copies with any tool you find appropriate

If you are going to use the native Magento backup function, navigate to System -> Tools -> Backups and perform System and Database backups



2. Log in to the Magento backend

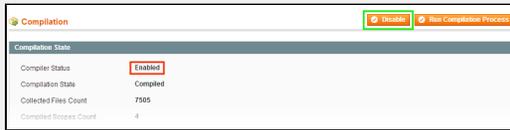
### 3. Disable compilation

#### Click to view details

Navigate to System -> Tools -> Compilation.

If Compiler status is Disabled, you can skip to the next step

If Compiler is enabled, disable it.



**IMPORTANT:** after the extension is installed, you can enable the compilation again; **IT IS CRUCIAL** that you use "Run Compilation Process" function, not just "Enable button"



**!** Installing an extension with the Compilation enabled will result in store downtime.

### 4. Flush store cache

#### Click to view details

You can flush the store cache in 2 ways:

- via the backend:

Navigate to System -> Cache Management menu, and click Flush Magento Cache button



- via filesystem:

On your server, navigate to Magento root folder, then proceed to /var/cache/; delete all the content there.

5. Download the extension package from your account and extract the downloaded archive

6. Copy the content of /Step\_1/ folder to your store's root directory



If you are using a custom theme,  
▼ [read the instructions here.](#)

The extension comes with 2 design packages: /base/default/ and /rwd/default/.

In case you are using a custom theme, it is recommended to copy the design files to your current theme's folders. In case there are several themes in use at the same store, the design files must be copied to each of them.

- Find this folder in the package: /step\_1/app/design/frontend/base/default/; copy its content to /app/design/frontend/[your\_package]/[your\_theme]/
- Find this folder in the package: /step\_1/skin/frontend/base/default/; copy its content to /skin/frontend/[your\_package]/[your\_theme]/

7. Copy the content of /Step\_2/ folder to your store's root directory

8. Flush store cache again; log out from the backend and log in again.

## Upgrading from version 1.x

If you have version 1.x installed, you have to wipe out the entries you've made during previous installation:

▼ [See details](#)

Open the app/design/frontend/default/default/layout/catalogsearch.xml file. If you use a different from the Default theme, please find and open this file from appropriate directory.

Remove the following line:

```
<block type="aw_searchautocomplete/view" name="top.search" as="topSearch"
template="searchautocomplete/view.phtml"/>
```

Clear Magento cache.

Note that CSS, JS, and image files were moved to their traditional Magento locations. Please make sure you have your custom CSS and image files in proper directories.

---

## How does it work?

Since version 3.4.8 Search Autocomplete and Suggest (SAAS) extension has the following workflow:

- 1) Each search query is split into separate words;
- 2) The extension then looks for EACH separate word in EACH product attribute that is relevant to search;
- 3) The results are built from the products that have ALL the search query words mentioned in ANY attributes (the words can be either in a single or different attributes);
- 4) Products on the results page are displayed according to the following pattern:
  - a) First go the products that have every word included in their name;

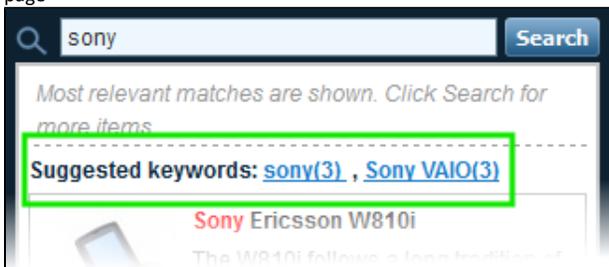
- b) Second come the products that have a single words in their name;
- 3) Third set lists the products of ALL other cases (sorting by product name).

## Configuration

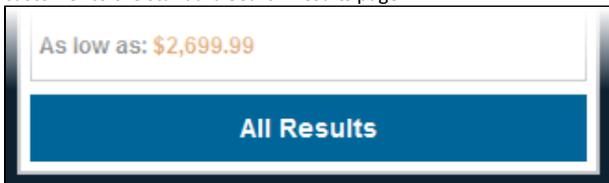
The extension's settings are available via System-> Configuration-> aheadWorks Extensions-> Search Autocomplete And Suggest-> Interface Settings.

### Basic representation settings

- Show products - this option defines how many items will be included in the dropdown list
- Show suggested keywords - enables / disables keyword suggestions. Clicking a keyword takes a customer to the standard Search Results page



- Show "All Result" button - enable/disables the correspondent button in the bottom of the dropdown. Clicking this button takes the customer to the standard Search Results page



- Show out of stock products - includes / excludes the items from the search result
- Open link in a new window - controls whether clicking dropdown items will open in a new or in the current window

Interface Settings		
Show products	<input type="text" value="3"/>	[STORE VIEW]
	<small>▲ Number of products to display</small>	
Show suggested keywords	<input type="text" value="Yes"/>	[STORE VIEW]
Show "All Result" button	<input type="text" value="Yes"/>	[STORE VIEW]
Show out of stock products	<input type="text" value="No"/>	[STORE VIEW]
Open link in a new window	<input type="text" value="No"/>	[STORE VIEW]

### Search item template

Search result item template field:

Here you can customize the appearance of the search item (i.e., a single product shown in the results). HTML is supported.

You may insert any product attribute of text and varchar type by using {attribut

e\_name} variable as shown below:

```
<strong>{name}</strong>
<p
style="font-size:0.9em;line-height:1.4em;">{short_descri
ption}</p>
```



Hint: since these attributes are available to be included in the search field list, you can find their names in the Search attributes multiselect field located below.

Please note that not all product attributes are meaningful in their text representation; moreover, some of them may be converted to a non-string type during loading, so the extension does not display such attributes.

The {product\_url} and {thumbnail\_url} variables pull the link to the product page and the thumbnail URL accordingly. You may use them to construct the link to the appropriate product page:

```
<a href="{product_url}"><strong>{name}</strong></a>
```

Additional options:

- Thumbnail width, px - defines the size of the product thumbnail
- Search results header - text displayed in the dropdown above the search results
- Search results footer - text displayed in the dropdown under the search results
- Nothing found notice - text displayed in the dropdown box if search query brought no results
- Preloader image - allows uploading a custom progress indicator

Search result item template	<pre>&lt;img src="{thumbnail_url}" style="float:left; margin:3px 8px 0 0;" /&gt; &lt;strong&gt;{name}&lt;/strong&gt; &lt;p style="font-size:0.9em;line- height:1.4em;margin:2px 0 0;"&gt;{short_description} &lt;/p&gt; &lt;strong&gt;{price}&lt;/strong&gt;</pre>	[STORE VIEW]
▲ All product attributes listed above are accessible by the {attribute_name} construction. Additional variables: {product_url} - link to product page {thumbnail_url} - product thumbnail URL {price} - product price		
Thumbnail width (px)	<input type="text" value="100"/>	[STORE VIEW]
Search results header	<input type="text" value="Most relevant matches are shown. Click Search for"/>	[STORE VIEW]
Search results footer	<input type="text"/>	[STORE VIEW]
Nothing found notice	<input type="text" value="Nothing found"/>	[STORE VIEW]
Preloader image	<input type="button" value="Choose File"/> No file chosen <input type="checkbox"/> Delete Image	[STORE VIEW]

## Search query configuration

- Query delay - defines the delay between the last input character and the start of search query.  
Note, the query can only start after 3 characters
- Search attributes - in this selector, you can select the product

attributes that extension will use for search.

Note, that attributes of the text and varchar types are available for search only.

- Use Advanced Search as search engine - (requires [Advanced Search](#) extension) defines whether the quick search will be carried out by the Search Autocomplete... own scripts, or by the Sphinx engine
- Search by tags - defines if the product tags will be included into searchable parameters

Query delay	<input type="text" value="500"/>	[STORE VIEW]
	<small>▲ In milliseconds, e.g. 500 means 0.5 of second</small>	
Search attributes	<div style="border: 1px solid #ccc; padding: 2px;"><ul style="list-style-type: none"><li>Activation Information (activation_information)</li><li>Allow Gift Message (gift_message_available)</li><li>Apply MAP (msrp_enabled)</li><li>Base Image (image)</li><li>Country of Manufacture (country_of_manufacture)</li><li>Country of Origin (country_origin)</li><li>Custom Design (custom_design)</li><li>Custom Layout Update (custom_layout_update)</li><li style="background-color: #f0f0f0;">Description (description)</li><li>Dimensions (dimension)</li></ul></div>	[STORE VIEW]
	<small>▲ Only attributes of text and varchar type are available</small>	
Use Advanced Search as search engine	<input type="text" value="No"/>	[STORE VIEW]
	<small>▲ This option requires <a href="#">aheadWorks Advanced Search extension</a> to be installed and configured properly.</small>	
Search by tags	<input type="text" value="Yes"/>	[STORE VIEW]

## Uninstallation

If you just need to temporary disable the extension, you can just do the steps 1-5, without deleting the files of the extension.

1. Disable compilation, in case it is enabled.
2. Login to your FTP, navigate to `app/etc/modules/`
3. Open the file `AW_Searchautocomplete.xml` and change the following line:

```
<active>true</active>
```

to

```
<active>>false</active>
```

Now your Magento is unaware of the existence of this module.

4. Clear the cache under `var/cache`
5. Make sure that the site is working properly, otherwise roll back the changes and apply to our [technical support](#).
6. If everything works fine, it is safe to delete the files of the extension.

## Troubleshooting

After the extension installation the store gives an error, or blank page, or suggests to start Magento installation procedure.

Change the owner of the extracted extension files to the web server user and set 775 permissions on them. Clear the store cache and try again.

After the extension installation the page System->Configuration->Search Autocomplete And Suggest returns 404 error.

Logout from backend and login back again.

There is no aheadWorks extensions under my configuration section, or having the extension tab clicked I get a blank page, or Access Denied error.

Clear the store cache, browser cookies, logout and login again.

I've set up everything correctly, inserted the HTML code but there is nothing on that page.

Clear the store cache, clear your browser cache and domain cookies and refresh the page.

My configuration changes do not appear on the store.

Clear the store cache, clear your browser cache and domain cookies and refresh the page.

---

You can always find the latest version of the software, full documentation, demos, screenshots and reviews at <http://ecommerce.aheadworks.com>

License agreement: <http://ecommerce.aheadworks.com/LICENSE-M1.txt>

Contact us: <http://ecommerce.aheadworks.com/contacts/>

Copyright © 2015 aheadWorks <http://www.aheadworks.com>