

# Cache Improve Extension Documentation by Innovo

## Extension Installation

### **If you acquired the extension from Magento Marketplace:**

Please follow instructions of installing extensions from

[http://docs.magento.com/marketplace/user\\_guide/quick-tour/install-extension.html](http://docs.magento.com/marketplace/user_guide/quick-tour/install-extension.html)

### **If you acquired the extension from our website [innovo.ro](http://innovo.ro):**

Step 1) We distribute the extension as an archive.

Unpack the extension on your computer disk.

Copy all unpacked contents to your server into the directory:

[your store magento 2 directory]/app/code/Innovo/CacheImprove/

Step 2) Compile again your Magento 2 code by running these commands on your server:

```
php bin/magento module:enable Innovo_CacheImprove
```

```
php bin/magento setup:upgrade
```

```
php bin/magento deploy:mode:set production
```

Step 3) Either go to Admin > System > Cache Management and click in Flush Cache Storage button or run the command:

```
php bin/magento cache:flush
```

## Admin configurations

Admin configurations are found in Admin > Stores > Configuration > Innovo > Cache Improve section.

**General > Active** option: Yes/No

Enables or disables extension functionalities.

**Cache Management > Clean Additional Cache Types:**

Clean additional cache types by entity tags before built-in Page Cache/Varnish is cleaned.

Recommendation: In Magento 2.1 set Block Html and Collections Data. In Magento 2.0 set Collections Data.

This behaviour is triggered after frontend checkout purchase is made while purchased products become out of stock.

**Cache Management > Force Add Product Tags:**

This options is meant to use in Magento 2.0.x installation where pages depend on categories cache tags instead of product cache tags.

This behaviour is triggered after fronted checkout purchase is made while purchased products become out of stock.

**Cache Management > Force Add Parent Product Tags:**

Recommended with Magento 2.1.3.

Force clean full page cache by parent product tags.

This behaviour is triggered after frontend checkout purchase is made while purchased products become out of stock.

**Top Menu > Apply Top Menu Cache Improvements:**

Enables or disables cache improvements applied by this extension.

If you're store has different top menu code other than Magento's default top menu, like top menu added from an extension, a theme or custom coded by your developers, than you can set this option to No.

**Top Menu > Top Menu block name(s):**

Top menu layout block names.

Format of this configurations is text separated: comma, space or new line.

Default top menu block name is "catalog.topnav".

**Top Menu > Remove ESI Policy:**

If set to 'Yes', top menu block will not use Varnish ESI policy anymore, such gaining extra server resources (usually above 50 – 100 ms).

Option best used in conjunction with 'Top Menu > Use Static Cache' set to 'Yes',

which we recommend to do.

**Top Menu > Override TTL ESI Policy value:**

Will display the option 'Top Menu > TTL ESI Policy value' and has meaning in extension's code too.

**Top Menu > TTL ESI Policy value:**

By default Magento 2 has this block TTL (time to live) set to 3600 seconds.

Please change this option as your store needs. We recommend to set higher values if your store menu look and contents does not change frequently during business hours.

Option depends on Top Menu > Remove ESI Policy is 'No' and 'Top Menu > Override TTL ESI Policy value' = 'Yes'.

**Top Menu > Use Static Cache:**

Decouples top menu cache from categories cache tags, such making it fast loading (server side) and insensitive to customer purchases.

**Note:** In Magento 2.0.x set this option to "Yes" to have the extension running properly.

**Recommendation:** Set this option to 'Yes' if your store's top menu contents do not change often during business hours.

**Top Menu > Cache Lifetime:**

New value of top menu block cache lifetime. The value is in seconds.

Default TTL is 3600.

**Recommendation:** If "Remove ESI Policy" is "No", set this value greater or equal than "TTL ESI Policy Value".

**Top Menu > Topmenu jQuery Selector Expressions:**

jQuery top menu selectors expressions.

Can set multiple selectors separated by new lines or pipe character.

When set to empty, active categories will not be marked.

Option used to mark active categories in top menu.

**Top Menu > jQuery Parent Items Selectors:**

jQuery top menu selectors expressions used to select parent items of current active item in top menu.

Can set multiple selectors separated by new lines or pipe character.

Option used to mark active categories in top menu.

**Top Menu > CSS Class of Active Item:**

CSS class to add to current active item in top menu.

Option used to mark active categories in top menu.

**Top Menu > CSS Class of Has Active Items:**

CSS class to add to all parent items of current active item.

Option used to mark active categories in top menu.

**Payment > Checkout Full Action Names (pre dispatch):**

This option acts as enabler of extension main feature on non standard checkout pages (like Paypal Express return page and other).

Potentially, the main feature of your extension will work, by configuring this option with 3<sup>rd</sup> party 'one step checkout' extensions or payment extensions.

Checkout full action names allowed to initialize the process of using cache on products still in stock.

Separated by new line or pipe character.

e.g 'paypal\_express\_placeOrder'.

'checkout\_index\_index' not included in this configuration.

Configuration hooked on pre dispatch action event.

**Payment > Checkout Full Action Names (post dispatch):**

This option acts as enabler of extension main feature on non standard checkout pages (like Paypal Express return page and other).

Potentially, the main feature of your extension will work, by configuring this option with 3<sup>rd</sup> party 'one step checkout' extensions or payment extensions.

Checkout full action names allowed to initialize the process of using cache on products still in stock.

Separated by new line or pipe character.

e.g 'paypal\_express\_returnAction'.

'checkout\_index\_index' not included in this configuration.

Configuration hooked on post dispatch action event.

**Inventory > Quantity Divisions:**

Clean cache when quantity divides with a number greater than 1.  
Leave empty to clean cache only when product gets out of stock.

**Inventory > Discrete Quantities:**

Clean cache when products quantities become these numbers.

Enter valid positive numbers separated by comma.

Can set intervals. eg. 1-5,21-30. First number in interval is included. Last number in interval is excluded.

Leave empty to clean cache only when product gets out of stock.

Support contact at [info@innovo.ro](mailto:info@innovo.ro)

**Thank you for purchasing our extension,  
Innovo team**