

# Google PageSpeed Optimize Manual

## Getting Started

Welcome to the **Google PageSpeed Optimizer Documentation**. Whether you are a new or an advanced user, you can find some useful information here.

First of all, we recommend you check the following link:

- [How to install extension](#)

## How to install the extension

1. Backup your store's database and web directory.
2. Login to your server's SSH console and navigate to the root directory of the Magento 2 store.
3. Copy the installation instructions from the page [My Downloadable Products](#) into the SSH console and press ENTER.
4. Run the command below to enable the extension:

```
php -f bin/magento module:enable Mirasvit_Core Mirasvit_Optimize Mirasvit_Extend
```

5. Run the command below to install the extension:

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

6. Run the command below to clean the cache:

```
php -f bin/magento cache:clean
```

7. Deploy static view files:

```
rm -rf pub/static/frontend/*; rm -rf pub/static/backend/*; rm -rf var/view_precompiled/*; php -f bin/magento setup:static-content:deploy
```

8. Make sure that the native Magento cronjob is configured and working correctly. The extension performs all tasks in cron.

## Google PageSpeed Optimizer settings

The settings of the Google PageSpeed Optimizer is grouped into the following sections:

- [Insight](#) - section for checking the pagespeed of your website.
- [JavaScript Optimization](#) - settings for JavaScript optimization.
- [CSS Optimization](#) - settings for CSS optimization.
- [Image Optimization](#) - settings for images optimization.
- [HTML Optimization](#) - settings for HTML optimization.

## Insight

This configuration section was designed for checking the page speed rate of your website for desktop and mobile versions.

The **Run GoogleSpeed Test** is used to run a quick speed test for your website's home page.

- **Configurations**
  - **URLs for monitoring** - specify your website's URLs that need to be monitored when determining page speed performance, placing each URL in a new line.

## JavaScript Optimization

- **Enable JavaScript optimization** - enable/disable JavaScript optimization for your website.  
Run `bin/magento setup:static-content:deploy` after saving the configurations.
- **Merge JavaScript files** - enable/disable merge JS files. This config depends on the default Magento "Merge JavaScript files" config that available only in developer mode (Stores - Configuration - Advanced - Developer - JavaScript Settings).  
If enabled, the extension will merge JavaScript files into one file to reduce the number of HTTP requests made by a browser during the page loading process.  
JavaScript bundling will not work when this option is enabled.
- **Minify JavaScript** - enable/disable minification JS files. This config depends on the default Magento "Minify JavaScript files" config that available only in developer mode (Stores - Configuration - Advanced - Developer - JavaScript Settings).  
If enabled, the extension combines JavaScript files into bundles when required for particular page types, excluding unnecessary bundle files from the page depending on the page type.  
Run `bin/magento setup:static-content:deploy` after saving the configurations.
- **Move JavaScript To Page Bottom** - enable/disable moving JS files to the bottom of the page. **How it works:** Before the HTML page sent to the browser, our module checks for all external JS resources (
- **Ignore URL List** - the list of pages (URL patterns), where JS will not move. Or a list of JS files (patterns for SCRIPT tags) that should not be moved.  
For example:
  - to exclude pages like `/category/page/*`, the pattern should be: `category/page`
  - to exclude JS files, there should be an added pattern for JS file name (if the JS file was added as an external resource, with the "src" attribute inside the SCRIPT tag), or part of the content inside the SCRIPT tag (if JS code added directly to the page)

Each pattern on the new line.

### Note

If default Magento JS merging is enabled, some JS files will not be excluded from moving to the bottom of the page if merged into one file.

Before adding patterns for JS file names, please check to see if these files are present on the page.

You can do this by searching in the source of the page. To open the source of your page in the browser, please press Ctrl + u (Windows) or ? + Option + u (Mac).

- **Defer YouTube videos** - enable/disable lazy load for YouTube videos.

If enabled, the extension will replace the YouTube video with the link-placeholder that leads to the original YouTube video. This option prevents a load of resources for YouTube videos during the page loading process.

## CSS Optimization

- **Merge CSS Files** - enable/disable merge CSS files. This config depends on the default Magento "Merge CSS files" config that available only in developer mode (Stores - Configuration - Advanced - Developer - CSS Settings).\ If enabled, the extension combines additional CSS files into a single file to reduce the number of HTTP requests.

Run `bin/magento setup:static-content:deploy` after saving the configurations.

- **Minify CSS Files** - enable/disable minification CSS files. This config depends on the default Magento "Minify CSS files" config that available only in developer mode (Stores - Configuration - Advanced - Developer - CSS Settings).

If enabled, the extension reduces the size of CSS files.

Run `bin/magento setup:static-content:deploy` after saving the configurations.

- **Move CSS To Page Bottom** - enable/disable moving CSS files to the bottom of the page. **How it works:** Before the HTML page sent to the browser, our module checks for all external CSS resources ( tags) in the HTML code. Then the extension moves all found resources to the bottom of the page, except those who match any of the exceptions from the "Do not move CSS that contains" config.\
- **Do not move CSS that contains** - list of patterns for the page's link tags, which should not be moved to the bottom of the page. (Available only if the **Move CSS To Page Bottom** option is enabled).
- **Defer Google Fonts** - enable or disable a loading defer of Google Fonts. If enabled, the page will display the default fonts and then apply Google Fonts after they have been loaded.

## Image Optimization

The extension uses a set of tools in optimizing or converting images. Images get optimized in the pub/media folder.

To check the status of tools, please run the command: `bin/magento mirasvit:optimize-image:validate`

### Note

Suppose you receive a message about some libraries that have not yet been installed. In that case, you can install them by running the command: `sudo yum install <library_name>` (Centos) or `sudo apt-get install <library_name>` (Ubuntu).

They can be installed all together, for example, by running: `sudo apt-get install jpegoptim optipng gifsicle webp imagemagick`

After which you can wait for a while (until optimization has been completed by cron) or else run commands for the optimization:

`bin/magento mirasvit:optimize-image:optimize` - Run images optimization process

`bin/magento mirasvit:optimize-image:webp` - Create a copy of the images in .webp format

The images optimization process doesn't require any additional commands in the crontab on your server. It runs by Magento's default crontab.

- **Use WebP images** - enable or disable, using webp image types instead of default image types. Some browsers like Safari, IE, and certain old versions of Firefox and Edge do not support WebP image format. For these browsers, default image types will be used. Clear the Magento cache after this option has been changed.
- **Image Quality level** - choose a percentage level of the picture compression (100% - without quality loss, 70% - acceptable to display on the site). This is applicable only for JPEG and GIF, and only compresses the original image (each new thread of compression will delete the previous one automatically). Compression can be done by cron or [CLI command](#).
- **Image Lazy Load** - allows the loading of images only if they appear in the browser's viewport.
  - **Enable Lazy Load** - enable or disable lazy loading of images. Clear the Magento cache after this option has been changed.
  - **Preload first N images** - the number of images which should be fully loaded when the page is still loading. (Available only if the **Enable Lazy Load** option is enabled).
  - **Do not use a lazy load for images** - list of images for which lazyload should not be used. Each pattern in the new line. (Available only if the **Enable Lazy Load** option is enabled).

## HTML Optimization

- **Minify HTML** - enable or disable HTML document minification. If enabled, the extension reduces the size of the loading HTML file. Run `bin/magento setup:static-content:deploy` after saving the configurations.

## Command Line Interface

Usage: `php -f bin/magento [options]`

- Run a Google PageSpeed test for the given page:

```
bin/magento mirasvit:optimize-insight:pagespeed https://example.com/
```

- Validate the software required for image optimization:

```
bin/magento mirasvit:optimize-image:validate
```

- Run the image optimization process:

```
bin/magento mirasvit:optimize-image:optimize
```

*Note: the Extension can also automatically optimize images in the cron*

- Run the restore command to the original image process:

```
bin/magento mirasvit:optimize-image:restore
```

*Note: the Extension can also automatically restore images in the cron, recommending that you set **Image Quality level = 100%** before running this command*

- Run the procedure for creating a copy of images in .webp format

```
bin/magento mirasvit:optimize-image:webp
```

*Note: the Extension can also automatically create .webp images in the cron*

## Troubleshooting

- [Magento Dashboard is not working](#)

### Magento Dashboard is not working after installing the extension

This happens on some servers. In order to fix it, run the command:

```
php bin/magento config:set mst_optimize/optimize_js/enabled 1 - to enable JS optimization from CLI.
```

## How to upgrade extension

To upgrade the extension, follow these steps:

1. Backup your store's database and web directory.
2. Login to your server's SSH console and navigate to the root directory of the Magento 2 store.
3. Run the command below to update the current extension with all dependencies:

```
composer require mirasvit/module-optimize:* --update-with-dependencies
```

#### Note

In some cases, the command above is not applicable; neither is it possible to update just the current module, nor need to upgrade all the Mirasvit modules in a bundle. In this case, the command above will be of no effect.

Run instead `composer update mirasvit/*` command. It will update all the Mirasvit modules installed in your store.

4. Run the command below to enable the extension:

```
php -f bin/magento module:enable Mirasvit_Core Mirasvit_Optimize Mirasvit_Webp
```

5. Run the command below to install updates:

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

6. Run the command below to clean the cache:

```
php -f bin/magento cache:clean
```

7. Deploy static view files:

```
rm -rf pub/static/frontend/*; rm -rf pub/static/backend/*; rm -rf var/
php -f bin/magento setup:static-content:deploy
```

## Disabling the Extension

### Temporarily Disable

To temporarily disable the extension please follow the following steps:

1. Login to your server's SSH console and navigate to the root directory of the Magento 2 store.
2. Run the command below to disable the extension:

```
php -f bin/magento module:disable Mirasvit_Optimize Mirasvit_ImageLazyload
```

3. Log in to the Magento backend and refresh the store's cache (if enabled).

### Removing the Extension

To uninstall the extension, please follow these steps:

1. Login to your server's SSH console and navigate to the root directory of the Magento 2 store.
2. Run the command below to remove the extension:

```
composer remove mirasvit/module-optimize
```

3. Log in to the Magento backend to refresh the store cache (if enabled).

## Change Log

### 1.0.18

(2020-10-19)

#### Fixed

- Fixed issue with adding preload fonts to ajax response ([#72]())
-

## **1.0.17**

*(2020-10-06)*

### **Improvements**

- Minor code changes
- 

## **1.0.16**

*(2020-10-01)*

### **Fixed**

- Compatibility with PHP 7.4
- 

## **1.0.14**

*(2020-09-16)*

### **Fixed**

- Fixed error on pages with a few YouTube videos ([#64]())
- 

## **1.0.13**

*(2020-09-15)*

### **Improvements**

- Validation for the ability to optimize images ([#62]())
- 

## **1.0.12**

*(2020-09-10)*

### **Improvement**

- Conflicts validation
-

## 1.0.11

(2020-09-07)

### Fixed

- Fixed issue with warnings during png images compression.
- 

## 1.0.10

(2020-09-04)

### Improvement

- Images compression ([#50]())
- 

## 1.0.9

(2020-09-01)

### Improvements

- Treshold for bundle/track requests ([#46]())

### Fixed

- Fixed issue with not all images displayed in some sliders ([#47]())
- 

## 1.0.8

(2020-08-27)

### Feature

- Lazy load for YouTube videos

### Improvements

- Merge JS files option in the configurations of the extension
- 

## 1.0.7



(2020-07-29)

## **Improvements**

- Support of Magento 2.4

## **Fixed**

- issue with CSS exceptions ([#37]())
- 

## **1.0.6**

(2020-06-17)

## **Fixed**

- Issue with lazy load for images with a path in single quotes
- 

## **1.0.5**

(2020-06-02)

## **Fixed**

- Issue with Magento 2.1.\* compatibility

## **Improvements**

- Disabling images conversion to webp when "Use WebP Images" is disabled
- 

## **1.0.4**

(2020-04-23)

## **Fixed**

- Not all images visible with webp and lazyload
  - Issue with webp conversion (Unsupported color conversion request)
- 

## **1.0.3**

(2020-03-10)

## **Improvements**

- PageSpeed Optimizer in admin menu
  - Move lazy-load exceptions to configuration
- 

## **1.0.2**

*(2020-03-05)*

## **Improvements**

- Ability to exclude JS files from movement to the bottom of page
  - Lazy-load (server transparent image placeholder in the original image size)
  - Request validation (post, ajax, checkout etc)
- 

## **1.0.1**

*(2020-02-10)*

## **Improvements**

- Additional exceptions (configurable in admin) for lazy-load

## **Fixed**

- Issue with css defer
  - Small config issues
- 

## **1.0.0**

*(2020-02-03)*

## **Improvements**

- Additional exceptions for lazy-load
- 

## **0.0.10**

*(2020-01-24)*

## **Improvements**

- Lazy load for webp images

- Ability to exclude specified URLs from move\_js

## **Fixed**

- Possible issues with measure google pagespeed score
- 

## **0.0.9**

*(2020-01-02)*

## **Improvements**

- WebP images settings
  - WebP Picture tag
  - Added the command for validate required image optimization software
- 

## **0.0.8**

*(2019-12-26)*

## **Improvements**

- WebP support
- 

## **0.0.7**

*(2019-12-25)*

## **Fixed**

- Issue with saving the settings
- 

## **0.0.6**

*(2019-12-24)*

## **Improvements**

- Lazy Images & Defer google fonts
- 

## **0.0.5**

*(2019-12-23)*

### **Fixed**

- Replace 1px empty image with transparent
- 

## **0.0.4**

*(2019-12-23)*

### **Improvements**

- Image optimization
  - Add css option to all css files: font-display:swap
  - Exception URL list for Move CSS option
  - Deffer CSS
- 

## **0.0.3**

*(2019-12-18)*

### **Improvements**

- Image optimization statistic
- JS minification
- PHPMD
- Debug option for lazy load images
- Optimize images queue

### **Fixed**

- Issue with LazyLoad (Firefox)
- 

## **0.0.2**

*(2019-12-16)*

### **Improvements**

- Image optimization module
- 

## **0.0.1**

*(2019-12-11)*

## **Improvements**

- Initial release