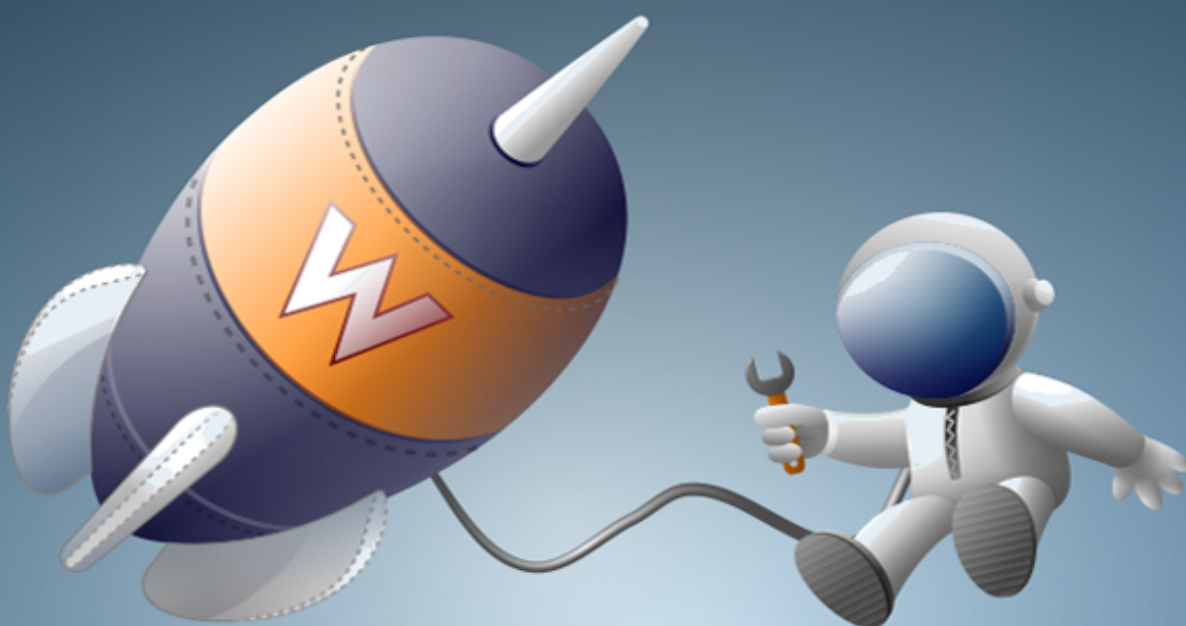




WEBO

Site SpeedUp

Руководство пользователя



Оглавление

1. Введение	4
Что такое WEBO Site SpeedUp?	5
Сравнение версий	7
Системные требования	8
2. Установка и обновление	10
Установка в качестве отдельного приложения.....	11
Установка плагина Bitrix.....	14
Установка плагина CS-Cart	17
Установка плагина Drupal	20
Установка плагина Joomla!	23
Установка плагина Magento	31
Установка плагина NetCat	34
Установка плагина UMI	37
Установка плагина WordPress	40
3. Использование WEBO Site SpeedUp	43
Панель управления	45
Настройка.....	48
Состояние системы	65
Персональные данные	69

4. Принципы работы WEBO Site SpeedUp.....	70
Интеграция с веб-сайтом.....	71
Алгоритм оптимизации	80
CSS-правила для лучшего создания CSS Sprites	83
Plugins API	86
Системные файлы.....	89
Алгоритм настройки	92
Шаблоны отложенной загрузки.....	98
5. Устранение неполадок и поддержка	101
Проблемы при установке	102
Проблемы на стороне клиента.....	105
Проблемы на стороне сервера.....	110
Техническая поддержка.....	114



1. Введение

Это руководство содержит информацию об установке и использовании приложения WEBO Site SpeedUp. Оно также поможет понять принципы работы всех используемых техник оптимизации, применяемых в WEBO Site SpeedUp.

1.1. Что такое WEBO Site SpeedUp?

WEBO Site SpeedUp автоматизирует все действия по улучшению клиентской составляющей веб-сайта, что приводит к существенному ускорению загрузки его страниц. Проверенные временем решения с открытым кодом вместе с уникальными инженерными находками объединяют все части процесса ускорения сайта, и теперь вы можете провести его буквально в два клика.

Действия по оптимизации

- Объединение внешнего и встроенного кода
- Минимизация файлов
- gzip-сжатие файлов
- Клиентское кэширование
- Серверное кэширование
- Поддержка множественных хостов
- Автоматическое создание CSS Sprites
- Автоматическое применение Data:URI
- Ненавязчивая загрузка JavaScript
- и многое другое...

Минимальное воздействие на быстродействие сервера

После того, как файлы кэша созданы, WEBO Site SpeedUp тратит около 1-5 мс на разбор и обновление HTML-документа (с настройками по умолчанию, опция вытягивания HTML-документа в одну строку — очень дорогостоящая процедура и может занимать дополнительно 50-100 мс).

Статистика ускорения

После ряда тестов на стандартных установках CMS WEBO Site SpeedUp показал среднее увеличение скорости загрузки страниц равное 250% от обычной скорости (и до 500% в некоторых случаях). Если быть точным, сайты ускоряются в среднем в 2,5 раза (+21 в оценке YSlow, -34% в размере, -43% в количестве внешних объектов).

Таблица ниже показывает насколько может увеличиться скорость загрузки веб-сайта после установки WEBO Site SpeedUp. Заметьте, что все CMS тестировались в идентичном окружении при пропускной способности канала в ~100КБ/с с помощью браузера Firefox 3.5. Веб-страницы загружались по несколько раз для того, чтобы кэш прокси-сервера был заполнен и для того, чтобы получить удовлетворительную статистику. Локальный кэш очищался после каждой загрузки.

CMS	YSlow		Время загрузки (с)		Размер (Кб)		Объекты	
	До	После	До	После	До	После	До	После
Bitrix 8.5.1	61	92	4.37	2.21	239	194	53	28
CMS Made Simple	74	96	0.483	0.375	49	35	15	5
cogear 1.0	76	91	5.11	1.57	447	129	11	9
DataLife Engine 8.0	65	91	4.48	1.29	147	120	43	16
Drupal 6.10	72	94	4.8	1.34	102	99	32	8
Etomite 1.1	89	96	0.874	0.477	19	14	7	6
ExpressionEngine 1.6.8	96	99	0.584	0.257	10	4	3	2
IPB 2.3.6	67	89	4.38	1.81	124	52	27	25
Joomla! 1.0.15	78	92	0.996	0.521	47	39	18	13
Joomla! 1.5.10	64	94	4.38	1.57	139	73	42	9
Joostina 1.2	63	91	8.07	3.77	426	333	45	17
Livestreet 0.3.1	51	96	10.87	1.97	298	111	48	5
MaxDev Pro 1.082	75	93	2.4	0.871	51	36	27	21
MaxSite 0.3.1	71	97	2.73	1.12	152	90	15	5
MODx 0.9.6.3	69	97	2.73	0.842	109	51	18	4
OpenSlaed 1.2	77	83	5.51	3.37	257	250	92	72
osCommerce 2.2	77	93	3.05	1.24	72	65	31	31
PHP-Nuke 8.0 *	72	91	2.785	1.272	181	91	19	19
phpBB 3.0.4	72	95	0.651	0.305	85	71	19	7
SMF 1.1.8 **	61	91	2.68	1.72	183	132	63	25
Textpattern 4.0.8	92	97	1.26	0.823	8	5	4	4
UMI.CMS 2.7	58	93	4.52	2.89	269	177	59	10
vBulletin 3.8.3	70	92	3.33	1.81	124	67	20	14
Website Baker 2.6	77	95	1.51	0.47	17	12	10	8
WordPress 2.7.1	72	95	4.58	2.08	133	125	31	6
Xaraya 1.1.5	81	97	1.79	0.78	35	16	8	4
XOOPS 2.3.3	72	95	3.22	1.53	65	50	21	8

* для PHP-Nuke был добавлен второй хост для статичных файлов

** для SMF 2 были добавлены два хоста для статичных файлов

1.2. Сравнение версий

WEBO Site SpeedUp существует в трех версиях: бесплатной, стандартной и расширенной версиях. Таблица ниже показывает различия между этими версиями.

Характеристики	Бесплатная версия	Стандартная версия	Расширенная версия
Ускорение первой и повторной загрузки страниц	до 200%	до 300%	до 500% и даже больше
Снижение нагрузки на сервер		есть	есть
Базовые возможности для ускорения	есть	есть	есть
Расширенные возможности для ускорения		есть	есть
Интеграция с CDN	Частично	Частично	Полностью
Расширенная SEO-совместимость			есть
Гарантия ускорения			есть
Установка и настройка продукта			есть
Отчет об ускорении			есть
Бесплатное обновление продукта	есть	есть	есть
Техническая поддержка		2 недели поддержки по e-mail	12 месяцев полной поддержки
Цена	Бесплатно	1799 рублей	Доступна только при заказе Ускорения Сайта

1.3. Системные требования

Серверные требования

- PHP 4.3+ или PHP 5+
- Apache или nginx или IIS server с модулем mod_php, или Denwer, или любой другой сервер с поддержкой CGI
- 16 Мб доступной памяти (64+ Мб желательно для корректного создания CSS Sprites)
- *Желательно:* наличие библиотеки curl (для загрузки внешних файлов)
- *Желательно:* наличие библиотеки zlib (для возможности gzip-сжатия при помощи PHP)
- *Желательно:* наличие библиотеки gd (для CSS Sprites)

Поддерживаемые браузеры

- IE 5+
- Firefox 2+
- Safari 2+
- Chrome
- Opera 7+

Поддерживаемые CMS

WEBO Site SpeedUp может быть установлен на *любую CMS* удовлетворяющую приведенным выше требованиям, но он также имеет встроенную поддержку следующих CMS:

- 4images
- Bitrix 8.5+ через системный плагин
- CakePHP 1.2.3+
- CMS Made Simple 1.6+
- CodeIgniter 1.7+
- cogear 1.0+
- Contao
- CS-Cart через системный плагин
- DataLife Engine 7.5+
- Drupal 5.x через системный плагин
- Drupal 6.x через системный плагин
- Drupal 7.x через системный плагин
- Etomite 1.1+
- ExpressionEngine 1.6.8+
- Geeklog 1.6+
- Invision Power Board 2.3.6+
- Joomla! 1.0.x

- Joomla! 1.5.x через системный плагин
- Joomla! 1.6.x через системный плагин
- Joomla! 1.7.x через системный плагин
- Joomla! 2.5.x через системный плагин
- Joostina 1.2+
- Kohana 2.3+
- Koobi CMS
- LiveStreet 0.2+
- Magento 1.2+
- Magento 1.4+ через системный плагин
- MaxDev Pro 1.082+
- MaxSite 0.3+
- MODx 0.9+
- NetCat
- NetCat 4.0+ через системный плагин
- Open Slaed 1.2+
- osCommerce 2.2+
- phpBB 3.0+
- PHP Fusion 7.0.5+
- PHP-Nuke 8.0+
- PrestaShop 1.2.5+
- Santafox 1.1+
- Simpla
- Simple Machines Forum 1.1.8+
- SocialEngine
- Symfony 1.2+
- Textpattern 4.0+
- TYPO3 4.2+
- TYPOlight
- VaM Shop 1.5.5+
- vBulletin 3.8.3+
- UMI.CMS 2.7+ через системный плагин
- WebAsyst
- Website Baker 2.8+
- WordPress 2.6+ через системный плагин
- X-Cart
- Xaraya 1.15+
- XOOPS 2.3.3+
- Yii 1.0+
- Zend Framework (с отключенным правилом перезаписи для WEBO Site SpeedUp)

Внимание: WEBO Site SpeedUp не работает на сайтах, не поддерживаемых через системные плагины и закодированных при помощи Zend.

2. Установка и обновление

Установка WEBO Site SpeedUp

WEBO Site SpeedUp просто установить на любой веб-сайт, отвечающий системным требованиям. По сути, это может быть любой веб-сайт с поддержкой PHP.

Существует два различных способа установки:

- Установка в качестве отдельного приложения. Это наиболее универсальный способ, подходящий для всех сайтов, отвечающих системным требованиям, и даже для тех, для которых возможна установка в качестве плагина.
- Установка в качестве плагина. WEBO Site SpeedUp может быть установлен в качестве плагина для WordPress, Joomla!, CS-Cart, Drupal или Bitrix. Установка в качестве плагина означает большую интеграцию и тем самым дополнительную производительность и кроме того, работу с WEBO Site SpeedUp в привычном интерфейсе панели администратора вашей системы.

В зависимости от выбранного вами способа установки WEBO Site SpeedUp выберите одну из детальных пошаговых инструкций, приведенных ниже:

- Установка в качестве отдельного приложения
- Установка плагина Bitrix
- Установка плагина CS-Cart
- Установка плагина Drupal
- Установка плагина Joomla!
- Установка плагина Magento
- Установка плагина NetCat
- Установка плагина UMI
- Установка плагина WordPress

Последняя версия приложения WEBO Site SpeedUp всегда может быть загружена с [официального сайта](#).

Обновление WEBO Site SpeedUp

WEBO Site SpeedUp сообщает о появлении новой версии приложения и отображает список изменений. Эта информация видна в блоке Обновления на странице Панель Управления, а также на вкладке Обновления на странице Состояние Системы в интерфейсе WEBO Site SpeedUp. Если вы заходите обновить WEBO Site SpeedUp, после одного клика по кнопке приложение автоматически загрузит все требуемые файлы обновления и установит их, сохраняя текущую конфигурацию.

Кроме того, если WEBO Site SpeedUp установлен в качестве плагина, все стандартные способы обновления плагинов также поддерживаются.

файлы и поддиректории (или как минимум убедитесь, что права на запись есть у файла `config.web.php` и директории `/cache/`). Это необходимо для нормальной работы приложения.

- Откройте в браузере следующую страницу: `http://your-website/web-optimizer/index.php`.

WEBO Site SpeedUp 0.9.6 Русский

Установка - ограничение доступа

Введите данные

Имя и фамилия:

Пароль:

* Подтверждение пароля:

Лицензионный ключ:

E-mail:

E-mail будет использован только для уведомления вас об экстренных обновлениях, поздравлений и специальных предложений.

Я прочитал [лицензионное соглашение](#) и ☒ согласен с ним:

Установить

Установка - ограничение доступа

Для обеспечения безопасности при дальнейшем использовании WEBO Site SpeedUp необходимо установить пароль доступа. **Некоммерческая версия** устанавливается без лицензионного ключа.

Перед проведением установки, пожалуйста, убедитесь, что корневой `.htaccess` и файлы используемой веб-системы доступны на запись (при проведении установки WEBO Site SpeedUp также создает резервные копии).


WEBO Site SpeedUp может самостоятельно проверить доступные возможности вашего сервера и провести установку в автоматическом режиме. Для этого нужно выбрать «Далее». В дальнейшем вы сможете изменить все сохраненные настройки, используя текущий интерфейс.

[Процесс установки WEBO Site SpeedUp](#)


- Введите *пароль*, *e-mail* и, в случае, если вы приобрели Облегченную или Полную версию WEBO Site SpeedUp, *лицензионный ключ*.
- Кликните по кнопке *Установить*. WEBO Site SpeedUp проанализирует окружение, установит исходную конфигурацию, а также произведет изменения необходимых файлов, сделав их резервные копии (как правило, это файлы `index.php` и `.htaccess`). Это займет лишь несколько секунд.
- Если ваша система не входит в поддерживаемых систем, найдите все файлы, отвечающие за вывод веб-страниц (как правило, это файл `index.php` и/или несколько других файлов `.php`). Каждый файл должен иметь несколько дополнительных строк в самом начале и в самом конце, следующим образом:

```
<?php require('/your-website-root/web-optimizer/
web.optimizer.php'); ?>
... прошлое содержимое файла ...
<?php $web_optimizer->finish(); ?>
```

- Установка завершена. Появится Панель Управления WEBO Site SpeedUp. Теперь вы можете изучить раздел *С чего начать*.


WEBO Site SpeedUp 0.9.6

Русский



Панель управления
Настройка
Состояние системы
Кэш
Персональные данные

Текущее состояние

WEBO Site SpeedUp **не работает** (режим отладки)

Вы можете отладить работу приложения:

- используя GET-параметр [web_optimizer_debug](#),
- или просто [при помощи cookies](#).

Нажмите кнопку «Включить», когда вы захотите перевести WEBO Site SpeedUp в рабочий режим.

[Включить](#)

Состояние кэша

CSS	3 Кб
JavaScript	9 Кб
Картинки	11 Кб
Общий размер	23 Кб

[Обновить кэш](#)

Закажите настройку

Возникли проблемы с настройкой WEBO Site SpeedUp?

Наши специалисты помогут установить и отладить WEBO Site SpeedUp для вашего сайта.

[Отправить заявку](#)

Настройка

Отключен Некоммерческая Облегченная **Полная**

0

Все в порядке

Инструменты оптимизации

- Статическое сжатие
- Оптимизация изображений

Доступно в полной версии

Обновления

Версия 0.9.7

- Improved UI (icons / sprites).
- Added option to exclude WEBO Site SpeedUp logic from website sections.
- Fixed CSS Tidy font property handling (used in data:URI / CSS Sprites).
- Improved smush.it callbacks.
- Improved static gzip tool.
- Improved CSS Sprites logic (images fetching).
- Improved gzip on CGI environments for CSS/JavaScript files.
- Fixed minor compatibility issues.

[Обновить до версии 0.9.7](#)

Быстрые ссылки

- [Сайт WEBO Site SpeedUp](#)
- [Пользовательская документация](#)
- [Известные проблемы](#)
- [Техническая поддержка](#)

Конфигурация сервера

- Нет поддержки [Symlinks](#)
- YUI Compressor не доступен
- Незащищенный режим работы

Расскажите о нас

[Joomla! Extensions Directory](#)

2.2. Установка модуля Bitrix

Важные замечания

- Если вы столкнетесь с какими-либо проблемами во время установки, пожалуйста, обратитесь к разделу Устранение неполадок и поддержка.
- Если вы используете модули PHP Speedy или Web Optimizer (или если они установлены как отдельные приложения), деактивируйте или удалите их перед установкой WEBO Site SpeedUp. Они не совместимы с WEBO Site SpeedUp.
- Если вы используете любые другие модули Bitrix для повышения производительности, рассмотрите возможность их отключения перед установкой и проверкой WEBO Site SpeedUp. Такие модули могут повлиять на работу WEBO Site SpeedUp. И заметьте, что WEBO Site SpeedUp объединяет в себе возможности многих подобных модулей, а в чем-то и превосходит их. Узнайте больше о [возможностях WEBO Site SpeedUp](#).
- WEBO Site SpeedUp не работает, когда вы авторизованы в Bitrix как Администратор. Чтобы убедиться, что WEBO Site SpeedUp работает, пожалуйста, выйдите из системы или используйте другой браузер или используйте режим отладки WEBO Site SpeedUp.
- После успешной установки, пожалуйста, обратитесь к разделу С чего начать.

Внимание: ниже описан процесс ручной установки. WEBO Site SpeedUp также может быть установлен как Стороннее обновление. [Подробнее об этом в блоге Bitrix](#).

Установка

1. Загрузите с [официального сайта](#) последнюю версию WEBO Site SpeedUp для Bitrix с именем вроде `webo.site.speedup.vx.x.x.bitrix.zip` (где X это цифры версии) и имеющий размер в несколько килобайт.
2. Извлеките этот архив в директорию `/bitrix/modules/webo.optimizer/`, туда, где установлен Bitrix. Должна получиться следующая структура:

```
/admin/  
/install/  
/lang/  
include.php  
LICENSE.txt  
options.php  
readme.txt
```

- В панели администратора Bitrix откройте *Панель Управления* → *Настройки* → *Модули* и нажмите кнопку Установить напротив модуля WEBO Site SpeedUp. WEBO Site SpeedUp загрузит необходимые файлы из репозитория, проанализирует окружение и установит исходную конфигурацию.

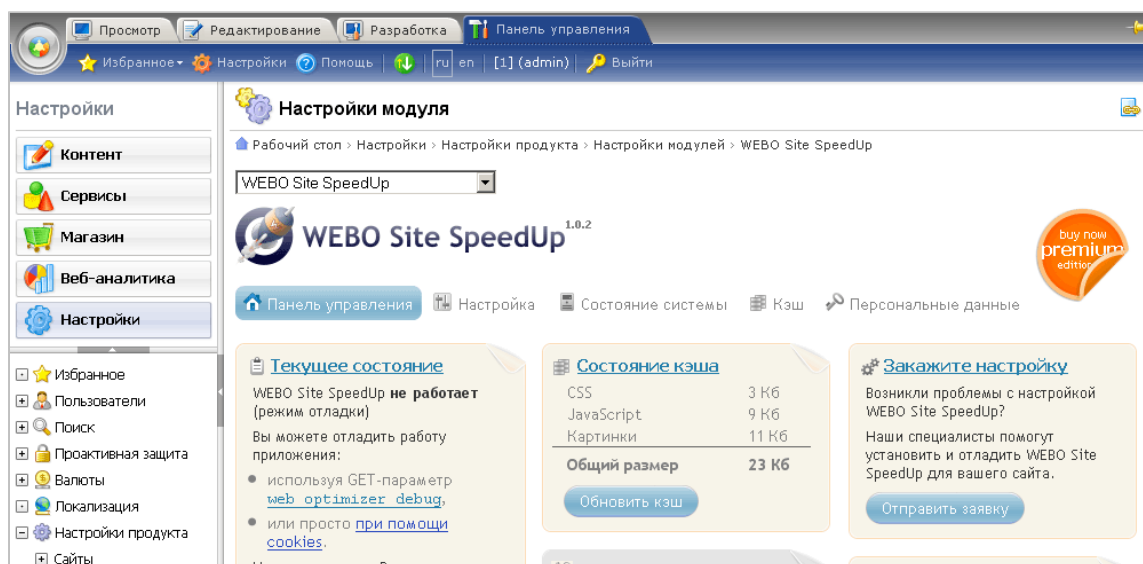
Пожалуйста, не предпринимайте никаких действий во время этого процесса, он может занять около минуты.

Название	Версия	Дата обновления	Статус	Действие
Главный модуль Ядро продукта с технологией "SiteUpdate".	8.5.1	05.10.2009	Установлен	
WEBO Site SpeedUp (Компания WEBO Software) WEBO Site SpeedUp обеспечивает комплексное ускорение сайта за счет применения всех технологий клиентской оптимизации	1.0.3	26.01.2010	Не установлен	Установить
Блоги Модуль дает возможность ведения блогов на сайте.	8.5.1	03.10.2009	Установлен	Удалить
Валюты Модуль управления валютами позволяет управлять валютами сайта и их курсами.	8.5.0	03.10.2009	Установлен	Удалить
Веб-аналитика Модуль сбора и отображения статистики сайта.	8.5.0	24.09.2009	Установлен	Удалить
Веб-сервисы Модуль позволяющий организовать систему веб-сервисов и SOAP.	8.5.0	19.06.2008	Установлен	Удалить
Веб-формы Модуль организует работу с произвольными веб-формами, позволяет хранить и фильтровать данные заполненных форм.	8.5.1	07.04.2009	Установлен	Удалить
Документооборот Модуль позволяющий организовать поэтапную обработку документов сайта.	8.5.0	24.09.2009	Установлен	Удалить
Интернет-магазин Позволяет осуществлять продажи товаров посетителям сайта.	8.5.0	22.09.2009	Установлен	Удалить
Информационные блоки Модуль работы с информационными блоками позволяет управлять и каталогизировать информацию различного характера - новости, вакансии, список продуктов.	8.5.1	05.10.2009	Установлен	Удалить
Компрессия Модуль компрессирующий страницы сайта для ускорения их загрузки.	8.5.0	03.10.2009	Установлен	Удалить

- В случае, если автоматическая установка невозможна, загрузите с [официального сайта](#) архив с необходимыми для WEBO Site SpeedUp файлами вручную. Архив должен называться примерно так: **webo.site.speedup.vx.x.x.zip** и должен иметь размер порядка 1 Мб. Извлеките этот архив в директорию **/bitrix/components/weboptimizer/**. Структура файлов в этой директории должна выглядеть примерно так:

```
/cache/
/controller/
...
config.webo.php
index.php
install.php
...
```

- Задайте права на запись для директории **/bitrix/cache/** включая все файлы и поддиректории и убедитесь, что файл **.htaccess** в корневой директории сайта также доступен для записи или, что он может быть создан, если он еще не существует (в этом случае корневая директория сайта должна быть доступна для записи). Это необходимо для нормальной работы модуля.
- Откройте **Control Panel → Settings → Modules Settings → WEBO Site SpeedUp** чтобы получить доступ к интерфейсу WEBO Site SpeedUp.



- Установка завершена. Появится сообщение об успешной активации модуля. Теперь вы можете изучить раздел С чего начать.

2.3. Установка плагина CS-Cart

Важные замечания

- Если вы столкнетесь с какими-либо проблемами во время установки, пожалуйста, обратитесь к разделу Устранение неполадок и поддержка.
- Если вы используете какие-либо модули CS-Cart для повышения производительности, рассмотрите возможность их отключения перед установкой и проверкой WEBO Site SpeedUp. Такие модули могут повлиять на работу WEBO Site SpeedUp. И заметьте, что WEBO Site SpeedUp объединяет в себе возможности многих подобных модулей, а в чем-то и превосходит их. Узнайте больше о [возможностях WEBO Site SpeedUp](#).
- После успешной установки, пожалуйста, обратитесь к разделу С чего начать.

Установка

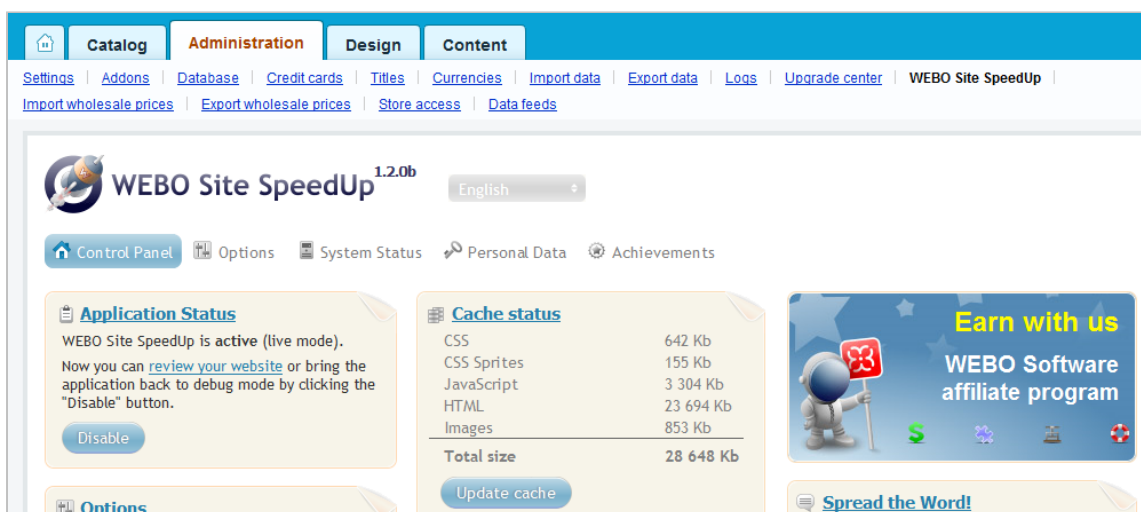
1. Загрузите с [официального сайта](#) последнюю версию WEBO Site SpeedUp для CS-Cart с именем вроде `webo.site.speedup.vx.x.x.cscart.zip` (где X это цифры версии). Размер файла должен составлять несколько килобайт.
2. Извлеките этот архив в директорию `/addons/`, туда, где установлен CS-Cart. В поддиректории `websitespeedup` должна получиться следующая структура:

```
/controllers/  
/schemas/  
addon.xml  
func.php  
init.php  
LICENSE.txt  
readme.txt
```

- Задайте права на запись для директории `/addons/websitespeedup/`, а также всех вложенных файлов и поддиректорий.
- Задайте права на запись для директории `/skins/имя_скина_панели_администратора/admin/addons/`.
- В панели администратора CS-Cart откройте *Administration* → *Addons* и напротив плагина WEBO Site SpeedUp plugin кликните по ссылке *Install*. CS-cart зарегистрирует плагин в системе.

Usergroup period	Active		edit	uninstall
Usergroup price	Active		edit	uninstall
Webmail			edit	install
WEBO Site SpeedUp			edit	install
Wg Alphabet Product Listing	Active		edit	uninstall
Wg Country Name Shipping	Active		edit	uninstall

- На той же странице смените статус плагина WEBO Site SpeedUp plugin с *Disabled* на *Active*. WEBO Site SpeedUp загрузит необходимые файлы из репозитория, проанализирует окружение и установит исходную конфигурацию. Пожалуйста, не предпринимайте никаких действий во время этого процесса, он может занять около минуты.
- Установка завершена. В панели администратора CS-Cart откройте *Administration* → *WEBO Site SpeedUp* чтобы увидеть интерфейс установленного плагина. Теперь вы можете изучить раздел *С чего начать*.



Устранение неполадок

Сообщение Страница не найдена на странице администрирования WEBO Site SpeedUp

Такое может происходить, когда часть файлов WEBO Site SpeedUp отсутствует в директории `/skins/basic/admin/addons/webositespeedup/`. Пожалуйста, убедитесь, что директория `/skins/basic/admin/addons/` доступна для записи (вместо `basic` подставьте название другого скина, если вы не используете стандартный). После этого попробуйте переустановить WEBO Site SpeedUp со страницы Аддонов.

Сообщение Возникла критическая ошибка на странице администрирования WEBO Site SpeedUp

Эта проблема проявляется тогда, когда какие-либо серверные модули по

соображениям безопасности блокируют динамические запросы WEBO Site SpeedUp к серверу. Чтобы избежать этой проблемы установите права в 755 и 644 для всех директорий и файлов, соответственно, в директории `/addons/webositespeedup/`, включая ее саму. Если это не поможет, постарайтесь получить текст сообщения об ошибке, возникающей при динамических запросах WEBO Site SpeedUp. Для этого вы можете использовать вкладку Консоль дополнения Firebug для Firefox.

Аддон активирован, но не отображается в основном меню

Возможно на сервере применяется кэширование. Попробуйте сбросить этот кэш одним из способов:

- Откройте ссылку `http://domain.com/cs-cart-admin.php?cc` (замените `domain.com` и `cs-cart-admin.php` на ваш домен и имя скрипта панели администратора, соответственно)
- Удалите все файлы из директории `/var/cache/`

2.4. Установка плагина Drupal

Важные замечания

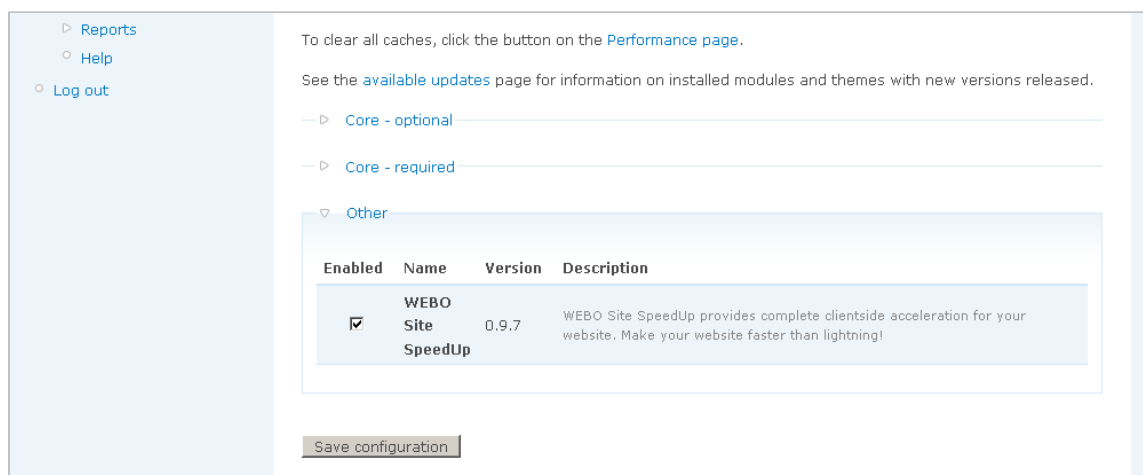
- Если вы столкнетесь с какими-либо проблемами во время установки, пожалуйста, обратитесь к разделу Устранение неполадок и поддержка.
- Если вы используете плагины PHP Speedy или Web Optimizer (или если они установлены как отдельные приложения), деактивируйте или удалите их перед установкой WEBO Site SpeedUp. Они не совместимы с WEBO Site SpeedUp.
- Если вы используете любые другие плагины Drupal для повышения производительности, рассмотрите возможность их отключения перед установкой и проверкой WEBO Site SpeedUp. Такие плагины могут повлиять на работу WEBO Site SpeedUp. И заметьте, что WEBO Site SpeedUp объединяет в себе возможности многих подобных плагинов, а в чем-то и превосходит их. Узнайте больше о [возможностях WEBO Site SpeedUp](#).
- Для достижения максимальной эффективности сжатия gzip пожалуйста отключите gzip в опциях Drupal. WEBO Site SpeedUp имеет лучшую поддержку gzip.
- После успешной установки, пожалуйста, обратитесь к разделу С чего начать.

Установка

1. Загрузите с [официального сайта](#) последнюю версию WEBO Site SpeedUp для Drupal с именем вроде `webo.site.speedup.vx.x.x.drupalx.zip` (где X это цифры версии) и имеющий размер в несколько килобайт.
2. Извлеките этот архив в директорию `/sites/all/modules/weboptimizer/`, туда, где установлен Drupal. Должна получиться следующая структура:

```
INSTALL.txt
LICENSE.txt
README.txt
weboptimizer.info
weboptimizer.module
```

- Установите права на запись для директории `/sites/all/modules/weboptimizer/` для веб-сервера (или для всех пользователей): WEBO Site SpeedUp автоматически установит сюда ядро продукта.
- В панели администратора Drupal откройте *Administer* → *Site building* → *Modules*, установите отметку в поле Enabled напротив модуля WEBO Site SpeedUp и кликните по кнопке *Save configuration*. WEBO Site SpeedUp загрузит необходимые файлы из репозитория, проанализирует окружение и установит исходную конфигурацию. Пожалуйста, не предпринимайте никаких действий во время этого процесса, он может занять около минуты. По окончании вы увидите сообщение о завершении операции.



- В случае, если автоматическая установка невозможна, загрузите с [официального сайта](#) архив с необходимыми для WEBO Site SpeedUp файлами вручную. Архив должен называться примерно так: `webo.site.speedup.vx.x.x.zip` и должен иметь размер порядка 1 Мб. Извлеките этот архив в ту же директорию `/sites/all/modules/weboptimizer/`. Теперь структура в этой директории должна выглядеть так:

```

/web-optimizer/
|__ /cache/
|__ /controller/
|__ ...
INSTALL.txt
LICENSE.txt
README.txt
weboptimizer.info
weboptimizer.module

```

- Задайте права на запись для директории `/web-optimizer/` включая все файлы и поддиректории или, как минимум, убедитесь, что права на запись есть у файла `config.webo.php` и директории `/cache/`. Это необходимо для нормальной работы плагина.
- Откройте *Administer* → *Site configuration* → *WEBO Site SpeedUp* чтобы получить доступ к интерфейсу WEBO Site SpeedUp.



- Установка завершена. Появится сообщение об успешной активации плагина. Теперь вы можете изучить раздел С чего начать.

2.5. Установка плагина Joomla!

Важные замечания

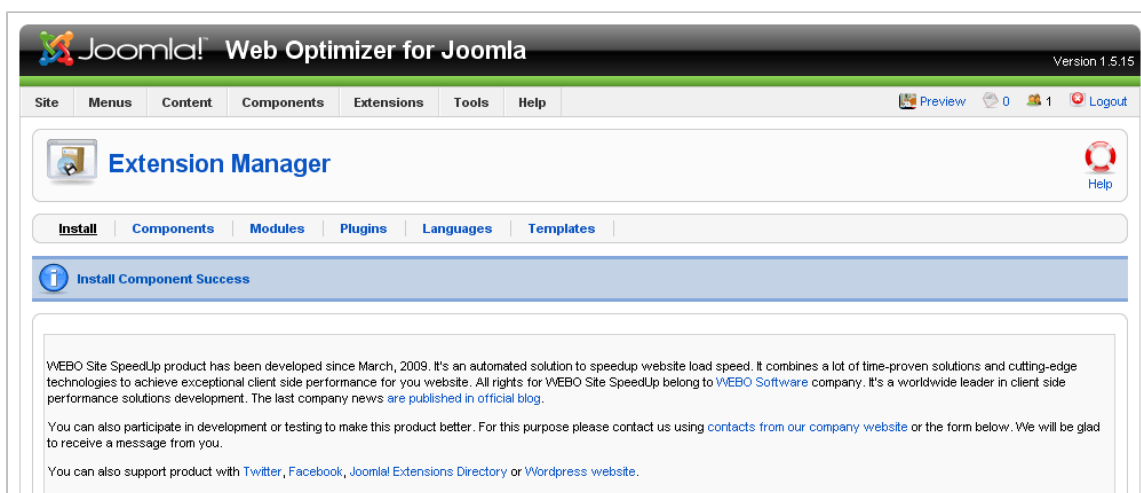
- Перед установкой, если вы уже используете PHP Speedy или Web Optimizer или другие средства для повышения производительности, отключите или удалите их. Такие средства могут отрицательно повлиять на работу WEBO Site SpeedUp. Заметьте, что WEBO Site SpeedUp объединяет в себе возможности многих подобных средств, а в чем-то и превосходит их. Узнайте больше о [возможностях WEBO Site SpeedUp](#).
- Если вы столкнетесь с какими-либо проблемами во время установки, пожалуйста, попробуйте найти их решение в разделе Устранение неполадок в конце этой статьи или обратитесь к общему разделу Устранение неполадок и поддержка.

Установка плагина Joomla! 1.5.x+

1. Откройте [официальный сайт WEBO Site SpeedUp](#) и найдите последнюю версию файла WEBO Site SpeedUp for Joomla! с именем вроде `webo.site.speedup.vx.x.x.joomlaхх.zip` (где X это цифры версии). Размер файла должен быть порядка 1 Мб. Загрузите этот файл или просто скопируйте его URL.
2. В панели администратора Joomla! откройте *Extensions* → *Install/Uninstall*.



3. В зависимости от первого шага, введите путь к загруженному файлу в поле *Upload Package* и нажмите кнопку *Upload & Install* ИЛИ введите путь к файлу на официальном сайте и нажмите кнопку *Install*. Начнется установка WEBO Site SpeedUp. Пожалуйста, не предпринимайте никаких действий во время этого процесса.



4. Появится сообщение об успешной установке компонента.



5. Установка завершена. В панели администратора Joomla! откройте *Components* → *WEBO Site SpeedUp* чтобы начать работу с плагином или обратитесь к разделу *С чего начать*, чтобы узнать об основных возможностях WEBO Site SpeedUp.

Установка плагина Joomla! 1.5.x+ вручную

- Загрузите с [официального сайта](#) последнюю версию WEBO Site SpeedUp для Joomla! с именем вроде `webositespeedup.vx.x.x.joomla15.zip` (где X это цифры версии). Размер файла должен составлять около 2 мегабайт
- Извлеките следующие файлы:

```
admin.webositespeedup.php
index.html
install.webositespeedup.php
LICENSE.txt
readme.txt
uninstall.webositespeedup.php
```



```
WeboMySQL.php
weboptimizer.php
weboptimizer.x
websitespeedup.xml
```

в директорию `/administrator/components/com_websitespeedup/`

- Извлеките директорию:

```
/web-optimizer/
```

в директорию `/components/com_websitespeedup/`

- Извлеките следующие файлы:

```
weboptimizer.php
websitespeedup.xml
```

в директорию `/plugins/system/`

- Извлеките файл:

```
WeboMySQL.php
```

в директорию `/libraries/joomla/database/database/`

- Выполните этот запрос к базе данных Joomla! (не забудьте заменить на используемый в вашей системе префикс):

```
INSERT INTO `[[dbprefix]]components` (`name`, `link`, `menuid`,
`parent`, `admin_menu_link`, `admin_menu_alt`, `option`,
`ordering`, `admin_menu_img`, `iscore`, `params`, `enabled`)
VALUES ('WEBO Site SpeedUp', 'option=com_websitespeedup', 0, 0,
'option=com_websitespeedup', 'WEBO Site SpeedUp',
'com_websitespeedup', 0, '../components/com_websitespeedup/web-
optimizer/favicon.ico', 0, '', 1);
INSERT INTO `[[dbprefix]]plugins` (`name`, `element`, `folder`,
`access`, `ordering`, `published`, `iscore`, `client_id`,
`checked_out`, `checked_out_time`, `params`)
VALUES ('System - WEBO Site SpeedUp', 'weboptimizer', 'system', 0,
2, 1, 0, 0, 0, '0000-00-00 00:00:00', '');
```

- **Опционально!** Измените значение параметра dbtype в файле `/configuration.php` (замените звездочку на предыдущее значение параметра dbtype)

```
var $dbtype = 'WeboMySQL';/*
```

Должно получиться примерно следующее:

```
var $dbtype = 'WeboMySQL';//mysqli
```

- Откройте файл `/components/com_websitespeedup/web-optimizer/config.webo.php` и установите следующие значения:

```
$compress_options['password'] = "[любой допустимый хэш пароля]";
$compress_options['host'] = "";
$compress_options['document_root'] = "";
$compress_options['website_root'] = "";
$compress_options['html_cachedir'] = "";
$compress_options['css_cachedir'] = "";
$compress_options['javascript_cachedir'] = "";
```

Должно получиться примерно следующее:

```
$compress_options['password'] =
"cbad86ef2bdfe8373bcade262354de45";
$compress_options['host'] = "site.com";
$compress_options['document_root'] = "/var/www/path-to-doc-root/";
$compress_options['website_root'] = "/var/www/path-to-doc-root/
joomla-installation/";
$compress_options['html_cachedir'] = "/var/www/path-to-doc-root/
joomla-installation/cache/wo/";
$compress_options['css_cachedir'] = "/var/www/path-to-doc-root/
joomla-installation/cache/wo/";
$compress_options['javascript_cachedir'] = "/var/www/path-to-doc-
root/joomla-installation/cache/wo/";
```

•

Установка завершена. Зайдите в панель администратора и откройте *Components* → *WEBO Site SpeedUp*, чтобы убедиться, что плагин установлен успешно.

Обновление с Joomla! 1.5.x до Joomla! 1.6.x/1.7.x

Для сохранения всех настроек WEBO Site SpeedUp перед обновлением Joomla! необходимо сделать следующее:

- Отключить WEBO Site SpeedUp в панели управления продукта.
- Обновить WEBO Site SpeedUp до последней стабильной версии на странице Состояние системы в продукте, чтобы сделать файлы конфигурации совместимыми с последней версией WEBO Site SpeedUp.
- Сделать резервные копии конфигурационных файлов:

```
/components/com_websitespeedup/web-optimizer/config.webo.php
/components/com_websitespeedup/web-optimizer/config.user.php
...
/components/com_websitespeedup/web-optimizer/config.userX.php
```

- Удалить WEBO Site SpeedUp из Joomla! (со страницы удаления компонентов).
- Обновить Joomla!
- Установить версию WEBO Site SpeedUp, совместимую с Joomla! 1.6/1.7.
- Обновить WEBO Site SpeedUp до последней стабильной версии (до той, под которой проводилось резервное копирование настроек) на странице Состояние системы.

- Скопировать и перезаписать все файлы настроек в ту же самую директорию:

```
/components/com_websitespeedup/web-optimizer/
```

- Проверить, что лицензионный ключ и все настройки компонента в порядке, и включить продукт в панели управления продукта.

Теперь все настройки WEBO Site SpeedUp успешно перенесены без какого-либо вреда для сайта.

Установка плагина Joomla! 1.0.x

WEBO Site SpeedUp не может быть установлен в качестве плагина в Joomla! 1.0.x., поэтому есть два пути:

- Установить последнюю версию WEBO Site SpeedUp [StandaloneInstallation в качестве отдельного приложения (рекомендуется).
- Загрузить и установить старую версию WEBO Site SpeedUp под названием Web Optimizer, которая поддерживает интеграцию с Joomla! 1.0.x в качестве Мамбота. Для того, чтобы это сделать:

#. Загрузите файл с именем web-optimizer.v0.6.7.joomla10.zip с [официального сайта](#).

#. В панели администратора Joomla! откройте *Installers* → *Mambot*.

#. Введите путь к загруженному файлу и нажмите кнопку *Upload & Install*.

#. Через несколько секунд появится сообщение об успешной установке.

#. Откройте директорию Joomla! и убедитесь, что файл `.htaccess` в этой директории доступен для записи (если он есть) и поддиректория `/cache/` доступна для записи включая все вложенные в нее файлы и директории. Это необходимо для нормальной работы плагина.

#. Установка завершена.

Удаление

1. В панели администратора Joomla! откройте *Components* → *WEBO Site SpeedUp* и нажмите кнопку *Отключить*.
2. Откройте *Extensions* → *Install/Uninstall* и выберите вкладку *Components*.
3. Найдите WEBO Site SpeedUp в списке компонентов, выберите его и кликните по иконке *Uninstall*.
4. Удаление завершено. Мы будем очень благодарны вам, если вы также сможете [сообщить нам](#) причину удаления WEBO Site SpeedUp. Мы всегда стремимся улучшить качество и эффективность наших продуктов.

Устранение неполадок

Установка прерывается с ошибкой `Component Install: Custom install routine failure` ИЛИ `Can't write to the file (please chmod it to 0664)` или другой похожей

Установите значение параметра *Enable FTP* на *No* на вкладке *Server* на странице *Site → Global Configuration*. Значение этого параметра может быть восстановлено после успешной установки WEBO Site SpeedUp.

Установка прерывается с ошибкой `unable to write` или ошибкой `Error when connecting WEBO Site SpeedUp SQL driver`

Убедитесь, что следующие директории доступны для записи:

- `/administrator/components/`
- `/cache/`
- `/components/`
- `/libraries/joomla/database/database/`

Это позволит Joomla! установить файлы WEBO Site SpeedUp, его директорию кэширования, и дополнительный драйвер баз данных (он необходим для функции кэширования SQL-запросов в WEBO Site SpeedUp в Joomla!).

Установка прерывается с ошибкой `Невозможно найти установочный пакет`

Такое может случаться, когда превышено ограничение на максимальный размер загружаемого файла в PHP. Если размер установочного пакета составляет 2 Мб, установите ограничение в 3 Мб, для примера, добавив следующую строку в файл `.htaccess` в корневой директории сайта:

```
php_value upload_max_filesize 3M
```

Помимо этого вы можете попробовать установить любую другую версию WEBO Site SpeedUp for Joomla! и после этого обновить ее до последней доступной версии используя панель администратора WEBO Site SpeedUp.

Сайт отображается нечитаемым или в виде белой (пустой) страницы

Отключите встроенную в Joomla! функцию сжатия, установив параметр *GZIP Page Compression* в *No* на странице *Site → Global Configuration*, на вкладке *Server*. WEBO Site SpeedUp тоже поддерживает сжатие с помощью `gzip` и в случае применения двойного сжатия будет возникать такая проблема.

Проблемы с внешним видом или неправильные пути к страницам/файлам *CSS/JavaScript/изображениям*

Убедитесь, что WEBO Site SpeedUp находится после плагина *System - SEF* и любых других плагинов *SEF*, но перед плагином *System - Cache* на странице

Extensions → *Plugin Manager*. В противном случае возможно появление неправильных URL файлов ресурсов.

Joomla! выдает ошибку `Fatal error: Class 'database' not found in /plugins/system/legacy.php`

Это происходит при включении Legacy Mode после установки WEBO Site SpeedUp, поскольку его функция кэширования SQL-запросов не совместима Legacy Mode. Чтобы избежать этой проблемы просто отключите плагин *System - Legacy* на странице *Extensions* → *Plugin Manager*. Помимо этого вы можете изменить параметр `dbtype` в файле `/configuration.php` на его оригинальное значение (см. ниже).

Joomla выдает белую (пустую) страницу или ошибку связанную с базой данных

Такая проблема может возникать тогда, когда SQL драйвер WEBO Site SpeedUp необходимый для кэширования SQL-запросов, несовместим с текущим окружением. Проблема может встречаться при использовании JoomFish. Смените значение параметра `dbtype` в файле `/configuration.php` в корневой директории Joomla! на его предыдущее значение. Вместо этой строки:

```
var $dbtype = 'WeboMySQL'; //mysql (или mysqli или любой другой
тип, который был ранее; он должен быть сохранен в этом
комментарии)
```

должна быть восстановлена оригинальная строка:

```
var $dbtype = mysql;
```

Значение этого параметра также можно изменить в поле *Database Type* на вкладке *Server* страницы *Site* → *Global Configuration*.

Нет следов работы WEBO Site SpeedUp

Отключите плагин *System - Cache* на странице *Extensions* → *Plugin Manager* и все другие кэширующие решения на вашем сайте. Такие решения усложняют настройку WEBO Site SpeedUp. Однако, по окончании настройки они при необходимости могут быть снова включены.

Не удастся отключить WEBO Site SpeedUp через панель администратора

1. Откройте файл `/components/com_websitespeedup/web-optimizer/config.webo.php` и измените значение параметра `$compress_options['active']` с "1" на "0".
2. Восстановите исходный файл `.htaccess` в корневой директории вашего сайта из файла `.htaccess.backup`, созданного WEBO Site SpeedUp.

Сравнение эффективности серверного кэширования в Joomla!

В таблице ниже приведено время генерации контента (HTML) при различной настройке кэширования

Кэш (настройка Joomla!)	System- Cache (плагин)	WEBO Site SpeedUp	Кэш WEBO Site SpeedUp	Экстрем. кэш WEBO Site SpeedUp	Время (мс)	YSlow
Нет	Нет	Нет	Нет	Нет	325	65
Да	Нет	Нет	Нет	Нет	143	65
Да	Нет	Да	Нет	Нет	152	93
Да	Да	Да	Нет	Нет	53	93
Нет	Нет	Да	Да	Нет	43	93
Нет	Нет	Да	Да	Да	4	93

2.6. Установка плагина Magento

Важные замечания

- Если вы столкнетесь с какими-либо проблемами во время установки, пожалуйста, обратитесь к разделу Устранение неполадок и поддержка.
- Если вы используете какие-либо модули Magento для повышения производительности, рассмотрите возможность их отключения перед установкой и проверкой WEBO Site SpeedUp. Такие модули могут повлиять на работу WEBO Site SpeedUp. И заметьте, что WEBO Site SpeedUp объединяет в себе возможности многих подобных модулей, а в чем-то и превосходит их. Узнайте больше о [возможностях WEBO Site SpeedUp](#).
- WEBO Site SpeedUp не работает, когда вы авторизованы в Magento как Администратор. Чтобы убедиться, что WEBO Site SpeedUp работает, пожалуйста, выйдите из системы или используйте другой браузер.
- После успешной установки, пожалуйста, обратитесь к разделу С чего начать.

Установка

1. Загрузите с [официального сайта](#) последнюю версию WEBO Site SpeedUp для Magento с именем вроде `webo.site.speedup.vx.x.x.magento.zip` (где X это цифры версии). Размер файла должен составлять около одного мегабайта.
2. Извлеките директории `'/Varien/'` и `'/Webo/'` из архива в директорию `/app/code/local/` установки Magento.
3. Убедитесь, что директория `/app/code/local/Webo/SiteSpeedup/web-optimizer/` доступна для записи, включая все файлы и поддиректории или, как минимум, убедитесь, что права на запись есть у файла `config.webo.php` и поддиректории `/cache/`.
4. Извлеките файл `'Webo_SiteSpeedup.xml'` из архива в директорию `/app/etc/modules/` установки Magento.
5. Установка завершена. В панели администратора Magento откройте *WEBO Site SpeedUp* чтобы увидеть интерфейс установленного плагина. Теперь вы можете изучить раздел С чего начать.

WEBO Site SpeedUp 1.2.1b

Global Record Search | Logged in as WEBO Software | Friday, July 2, 2010 | [Log Out](#)

Dashboard | Sales | Catalog | Customers | Promotions | Newsletter | CMS | Reports | System | **Web Site SpeedUp** | [Get help for this page](#)

Control Panel | Options | System Status | Personal Data | Achievements

Application Status
WEBO Site SpeedUp is active (live mode).
Now you can [review your website](#) or bring the application back to debug mode by clicking the "Disable" button.
[Disable](#)

Cache status
CSS: 11 Kb
JavaScript: 19 Kb
Images: 912 Kb
Total size: 942 Kb
[Refresh cache](#)

Speed up website and get WEBO Site SpeedUp for free

Options
Barely | Normally | Fast | Flying
44
+5 Apply CSS Sprites
+5 Distribute images
+5 Do not use regular expressions
[Share results on Twitter](#)

Optimization Results
Acceleration gained: 27.31 s → 4.86 s **462%**
Savings: 181 Kb → 86 Kb **53%**
[Share results on Twitter](#)

Current achievements
Options: 44
WEBO grade: 83
Files: 7
Website size: 86 Kb
Speedup: 462%
[Share results on Twitter](#)

[Spread the Word!](#)

Available Blocks

Устранение неполадок

После установки WEBO Site SpeedUp не появился в панели администратора

1. В панели администратора Magento откройте *System* → *Cache Management* и убедитесь, что отключен кэш Configuration или кликните по кнопке *Flush Magento Cache*, чтобы обновить этот кэш и WEBO Site SpeedUp был обнаружен системой.

Cache Storage Management [Flush Magento Cache](#) [Flush Cache Storage](#)

Select All | Unselect All | Select Visible | Unselect Visible | 0 items selected | Actions Refresh Submit

Cache Type	Description	Associated Tags	Status
<input type="checkbox"/> Configuration	System(config.xml, local.xml) and modules configuration files(config.xml).	CONFIG	ENABLED
<input type="checkbox"/> Layouts	Layout building instructions.	LAYOUT_GENERAL_CACHE_TAG	ENABLED
<input type="checkbox"/> Blocks HTML output	Page blocks HTML.	BLOCK_HTML	ENABLED
<input type="checkbox"/> Translations	Translation files.	TRANSLATE	ENABLED
<input type="checkbox"/> Collections Data	Collection data files.	COLLECTION_DATA	ENABLED
<input type="checkbox"/> EAV types and attributes	Entity types declaration cache.	EAV	ENABLED
<input type="checkbox"/> Web Services Configuration	Web Services definition files (api.xml).	CONFIG_API	ENABLED

Additional Cache Management

[Flush Catalog Images Cache](#) Pregenerated product images files.

[Flush JavaScript/CSS Cache](#) Themes JavaScript and CSS files combined to one file.

2. В панели администратора Magento откройте *System* → *Configuration*, после чего откройте ссылку *Advanced* в меню слева. Убедитесь, что вывод для WEBO Site SpeedUp разрешен. Если это не так, измените значение на *Enable*.

SERVICES	Mage_Sendfriend	Enable
Magento Core Api	Mage_Shipping	Enable
ADVANCED	Mage_Sitemap	Enable
Admin	Mage_Tag	Enable
System	Mage_Tax	Enable
Advanced	Mage_Usa	Enable
Developer	Mage_Wishlist	Enable
	WebSiteSpeedup	Enable

Некоторые файлы CSS и JavaScript уже объединяются

Для наиболее эффективного ускорения желательно отключить встроенные возможности Magento по объединению файлов. Для этого нужно открыть страницу *System* → *Configuration* → *Advanced* → *Developer*, после чего в блоках *CSS Settings* и *JavaScript Settings* отключить объединение файлов и сохранить конфигурацию.

Удаление

1. Отключите WEBO Site SpeedUp в его панели управления. Если панель управления по какой-либо причине недоступна, откройте файл `/app/code/local/Webo/SiteSpeedup/web-optimizer/config.web.php` и измените значение `$compress_options['active']` с "1" на "0". Помимо этого, если вы отключили WEBO Site SpeedUp вручную, изменив конфигурационный файл, вы можете восстановить исходный файл `.htaccess` в корневой директории сайта из файла `.htaccess.backup` созданного WEBO Site SpeedUp.
2. Удалите директории `/varien/` и `/web/`, находящиеся в директории `/app/code/local/`.
3. Удалите файл `webSiteSpeedup.xml`, находящийся в директории `/app/etc/modules/`.
4. Проверьте, что кэш Configuration сброшен.
5. Удаление завершено. Мы будем очень благодарны вам, если вы также сможете [сообщить нам](#) причину удаления WEBO Site SpeedUp. Мы всегда стремимся улучшить качество и эффективность наших продуктов.

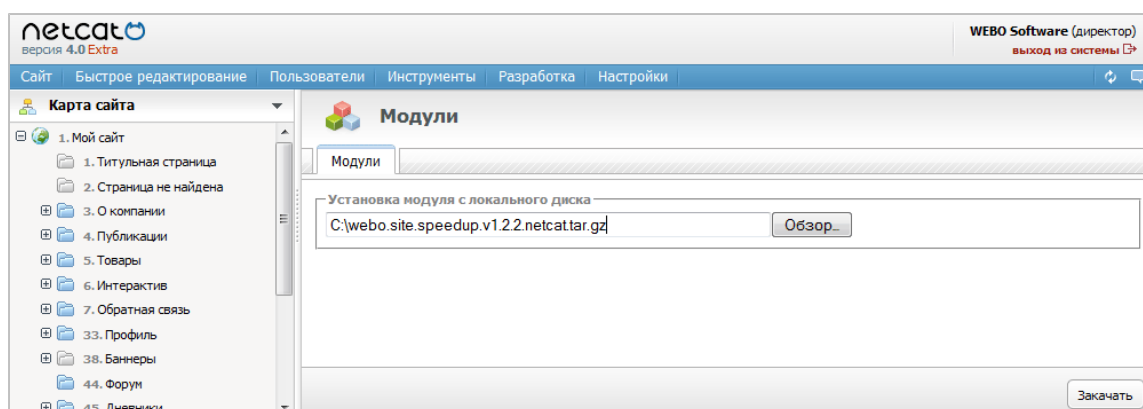
2.7. Установка плагина NetCat

Важные замечания

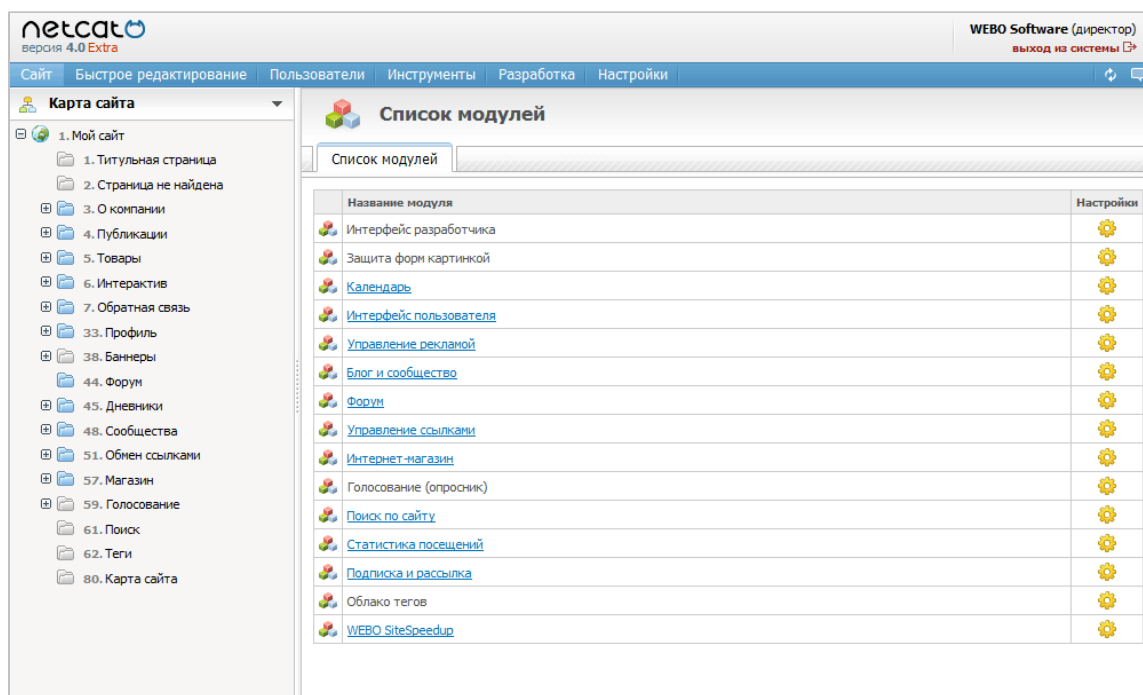
- Если вы столкнетесь с какими-либо проблемами во время установки, пожалуйста, обратитесь к разделу Устранение неполадок и поддержка.
- Если вы используете какие-либо модули Magento для повышения производительности, рассмотрите возможность их отключения перед установкой и проверкой WEBO Site SpeedUp. Такие модули могут повлиять на работу WEBO Site SpeedUp. И заметьте, что WEBO Site SpeedUp объединяет в себе возможности многих подобных модулей, а в чем-то и превосходит их. Узнайте больше о [возможностях WEBO Site SpeedUp](#).
- WEBO Site SpeedUp в качестве системного плагина может быть установлен только на NetCat версии 4.0 и выше. Если вы используете более раннюю версию NetCat, обратитесь к разделу Установка в качестве отдельного приложения.
- После успешной установки, пожалуйста, обратитесь к разделу С чего начать.

Установка

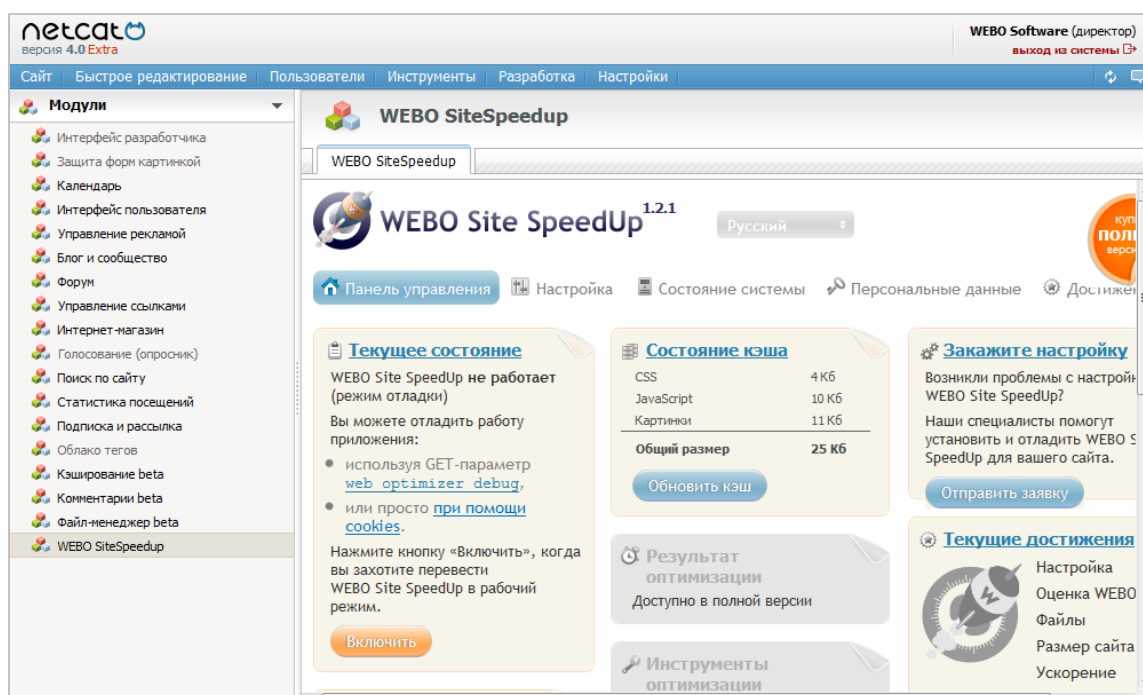
1. Загрузите с [официального сайта](#) последнюю версию WEBO Site SpeedUp для NetCat с именем вроде `webo.site.speedup.vx.x.x.netcat.tar.gz` (где X это цифры версии). Размер файла должен составлять несколько килобайт.
2. В панели администратора NetCat *Инструменты* → *Установка модуля*. В поле *Установка модуля с локального диска* укажите путь к загруженному файлу и нажмите кнопку *Закачать*. WEBO Site SpeedUp загрузит необходимые файлы из репозитория, проанализирует окружение и установит исходную конфигурацию. Пожалуйста, не предпринимайте никаких действий во время этого процесса, он может занять около минуты.



3. Откройте *Настройки* → *Список модулей*. Вы должны увидеть установленный модуль WEBO Site SpeedUp в списке модулей.



- Установка завершена. В списке модулей кликните по строке **WEBO Site SpeedUp**, чтобы перейти к панели администратора WEBO Site SpeedUp. Теперь вы можете изучить раздел **С чего начать**.



Удаление

- Отключите WEBO Site SpeedUp в его панели управления. Если панель управления по какой-либо причине недоступна, откройте файл `/app/code/local/Webo/SiteSpeedup/web-optimizer/config.web.php` и ИЗМЕНИТЕ

значение `$compress_options['active']` с "1" на "0". Помимо этого, если вы отключили WEBO Site SpeedUp вручную, изменив конфигурационный файл, вы можете восстановить исходный файл `.htaccess` в корневой директории сайта из файла `.htaccess.backup` созданного WEBO Site SpeedUp.

2. Удалите модуль стандартным для NetCat способом.
3. Удаление завершено. Мы будем очень благодарны вам, если вы также сможете [сообщить нам](#) причину удаления WEBO Site SpeedUp. Мы всегда стремимся улучшить качество и эффективность наших продуктов.

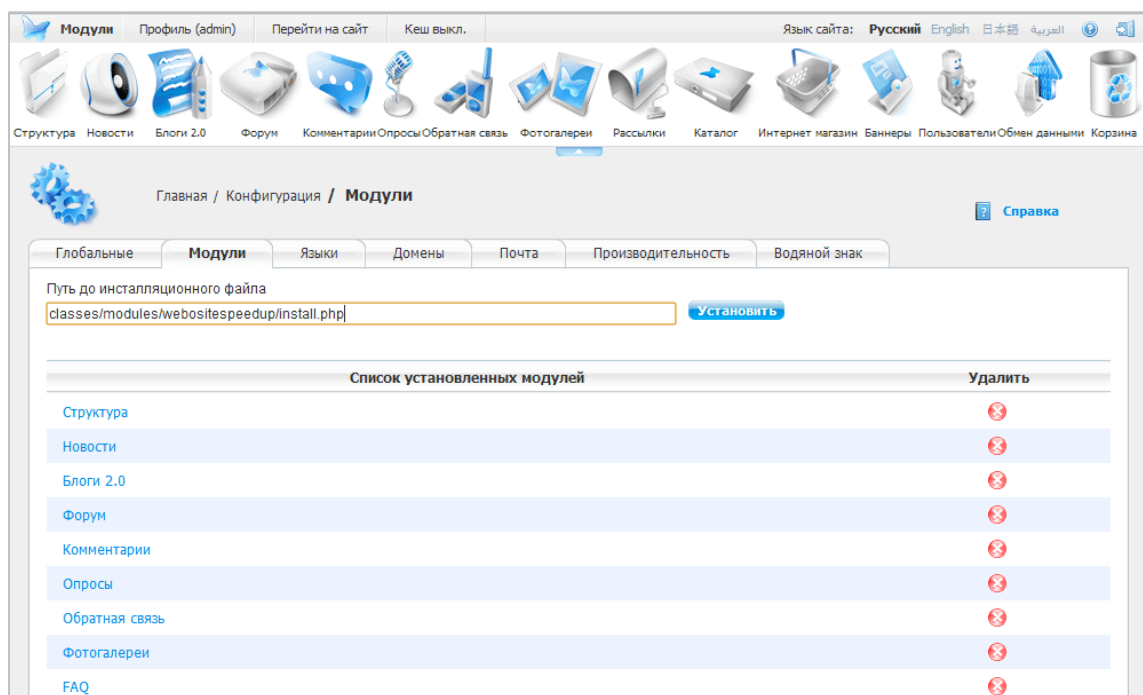
2.8. Установка плагина UMI

Важные замечания

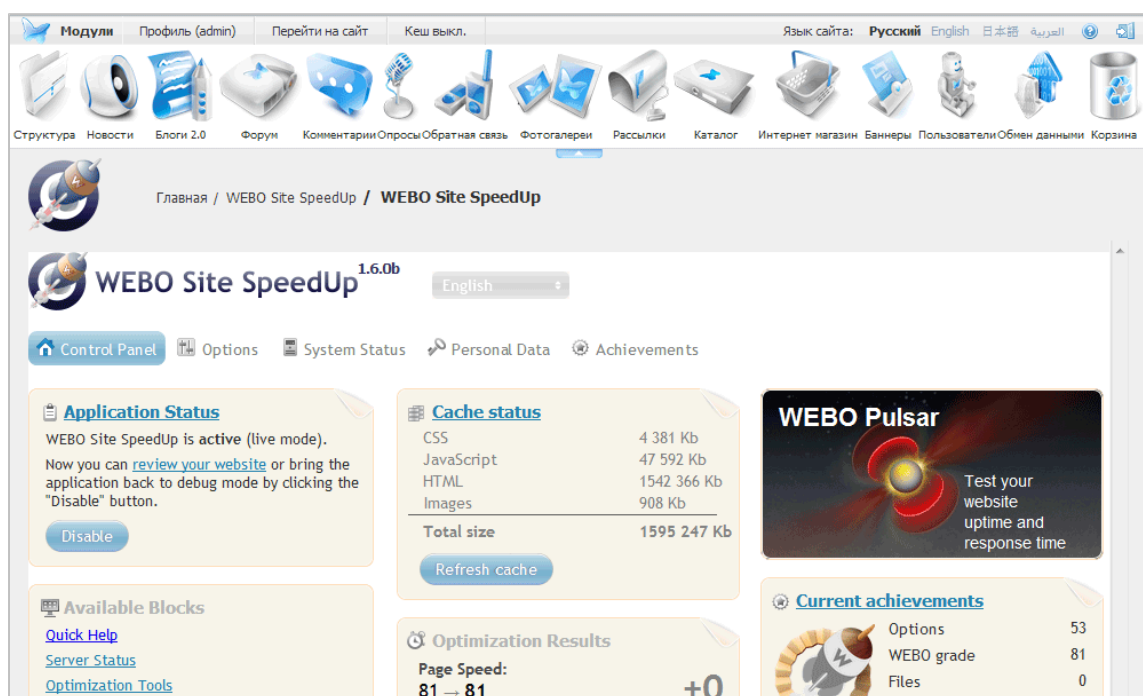
- Если вы столкнетесь с какими-либо проблемами во время установки, пожалуйста, обратитесь к разделу Устранение неполадок и поддержка.
- Если вы используете какие-либо модули Magento для повышения производительности, рассмотрите возможность их отключения перед установкой и проверкой WEBO Site SpeedUp. Такие модули могут повлиять на работу WEBO Site SpeedUp. И заметьте, что WEBO Site SpeedUp объединяет в себе возможности многих подобных модулей, а в чем-то и превосходит их. Узнайте больше о [возможностях WEBO Site SpeedUp](#).
- После успешной установки, пожалуйста, обратитесь к разделу С чего начать.

Установка

1. Загрузите с [официального сайта](#) последнюю версию WEBO Site SpeedUp для UMI с именем вроде `webo.site.speedup.vx.x.x.umi.zip` (где X это цифры версии). Размер файла должен составлять несколько килобайт.
2. Извлеките директорию содержимое архива в директорию `/classes/modules/webositespeedup/` установки UMI и установите права на запись для веб-сервера на эту директорию.
3. В панели администратора UMI откройте *Модули* → *Конфигурация* → *Модули*. В поле *Путь до инсталляционного файла* укажите путь `'classes/modules/webositespeedup/install.php'` и нажмите кнопку *Установить*. WEBO Site SpeedUp загрузит необходимые файлы из репозитория, проанализирует окружение и установит исходную конфигурацию. Пожалуйста, не предпринимайте никаких действий во время этого процесса, он может занять около минуты.



4. После окончания установки WEBO Site SpeedUp появится в списке установленных модулей на этой странице.
5. Установка завершена. Откройте страницу *Модули* → *WEBO Site SpeedUp* чтобы перейти к панели администратора WEBO Site SpeedUp. Теперь вы можете изучить раздел *С чего начать*.



Удаление

1. Отключите WEBO Site SpeedUp в его панели управления. Если панель

управления по какой-либо причине недоступна, откройте файл `/app/code/local/Webo/SiteSpeedup/web-optimizer/config.webo.php` и измените значение `$compress_options['active']` с "1" на "0". Помимо этого, если вы отключили WEBO Site SpeedUp вручную, изменив конфигурационный файл, вы можете восстановить исходный файл `.htaccess` в корневой директории сайта из файла `.htaccess.backup` созданного WEBO Site SpeedUp.

2. Удалите модуль стандартным для UMI способом.
3. Удаление завершено. Мы будем очень благодарны вам, если вы также сможете [сообщить нам](#) причину удаления WEBO Site SpeedUp. Мы всегда стремимся улучшить качество и эффективность наших продуктов.

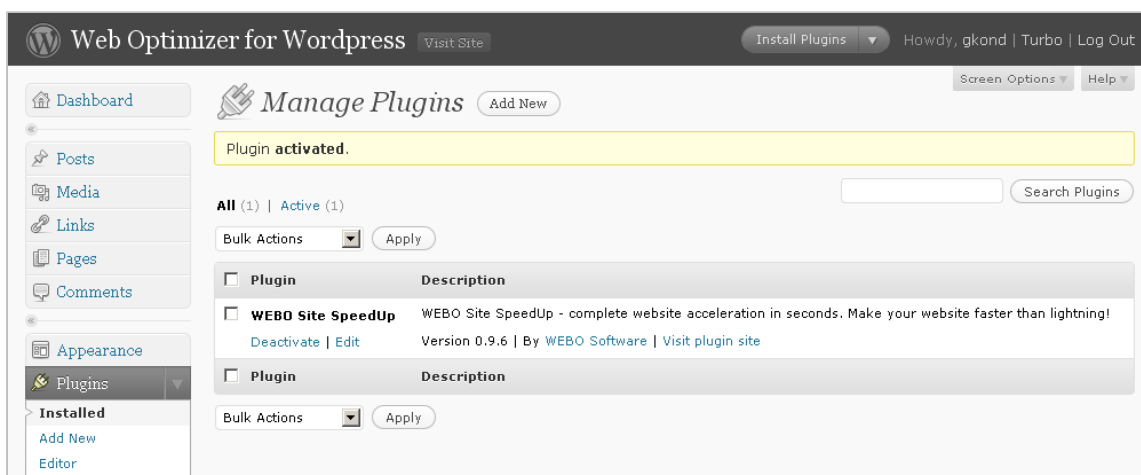
2.9. Установка плагина WordPress

Важные замечания

- Если вы столкнетесь с какими-либо проблемами во время установки, пожалуйста, обратитесь к разделу Устранение неполадок и поддержка.
- Если вы используете плагины PHP Speedy или Web Optimizer (или если они установлены как отдельные приложения), деактивируйте или удалите их перед установкой WEBO Site SpeedUp. Они не совместимы с WEBO Site SpeedUp.
- Если вы используете любые другие плагины WordPress для повышения производительности, рассмотрите возможность их отключения перед установкой и проверкой WEBO Site SpeedUp. Такие плагины могут повлиять на работу WEBO Site SpeedUp. И заметьте, что WEBO Site SpeedUp объединяет в себе возможности многих подобных плагинов, а в чем-то и превосходит их. Узнайте больше о [возможностях WEBO Site SpeedUp](#).
- После успешной установки, пожалуйста, обратитесь к разделу С чего начать.

Установка

1. Загрузите с [официального сайта](#) последнюю версию плагина WEBO Site SpeedUp для WordPress с именем вроде `webo.site.speedup.vx.x.x.wordpress.zip` (где X это цифры версии) и размером в несколько килобайт.
2. Извлеките архив в директорию `/wp-content/plugins/webo-site-speedup/`, туда, где установлен WordPress.
3. В панели администратора WordPress кликните *Плагины* → *Установленные* (*Plugins* → *Installed*).
4. Найдите в списке плагинов WEBO Site SpeedUp и кликните *Активировать* (*Activate*). WEBO Site SpeedUp загрузит необходимые файлы из репозитория, проанализирует окружение и установит исходную конфигурацию. Пожалуйста, не предпринимайте никаких действий во время этого процесса, он может занять около минуты. По окончании вы увидите сообщение об успешной активации плагина.



5. Задайте права на запись для директории `/webo-site-speedup/` включая все файлы и поддиректории или, как минимум, убедитесь, что права на запись есть у файла `config.webo.php` и поддиректории `/cache/`. Это необходимо для нормальной работы плагина.
6. Установка завершена. Теперь вы можете изучить раздел [С чего начать](#).

Устранение неполадок

Сообщение об ошибке `Curl isn't installed` во время установки WEBO Site SpeedUp

Если на вашем сервере отсутствует расширение PHP cURL плагин не может автоматически загрузить необходимые файлы из репозитория. Чтобы завершить установку необходимо выполнить два простых шага:

- Загрузить полный архив плагина (файл вида `webo.site.speedup.vx.x.x.zip`) из [официального репозитория](#).
- Извлечь содержимое директории `web-optimizer` из загруженного архива в ту же директорию, где уже находятся остальные файлы плагина: `/wp-content/plugins/webo-site-speedup/`.

Обновление с Web Optimizer для WordPress

1. Деактивируйте плагин Web Optimizer.
2. Загрузите WEBO Site SpeedUp поверх него или просто обновите плагин при помощи панели администратора WordPress.
3. Удалите директорию `/wp-content/plugins/web-optimizer/`.
4. Активируйте плагин. Все необходимые файлы будут загружены из репозитория автоматически.
5. После активации, пожалуйста, откройте страницу настройки WEBO Site SpeedUp и перейдите к странице **Состояние Системы**, на вкладку **Обновление**. После этого выберите флаг **Показывать информацию о бета-версиях**.

6. Обновите плагин до последней бета-версии. Конфигурационный файл будет автоматически изменен в соответствии с новой версией плагина.
7. Удалите файл `/wp-content/plugins/web.optimizer.wordpress.config.php`.
8. Теперь все обновления плагина могут быть безопасно установлены при помощи панели администратора WordPress или при помощи самого плагина WEBO Site SpeedUp.

3. Использование WEBO Site SpeedUp

Авторизация В отдельном приложении

Панель администратора WEBO Site SpeedUp, установленного в качестве отдельного приложения находится по тому же адресу, который используется для установки (должен быть примерно таким: <http://your-website.com/web-optimizer/>). Для авторизации необходимо указать пароль, заданный при установке WEBO Site SpeedUp. Если вы забыли пароль, пожалуйста, обратитесь к разделу Проблемы при установке.

Авторизация В плагине

В случае, если WEBO Site SpeedUp установлен в качестве плагина, его настройка осуществляется при помощи привычного интерфейса панели администратора этих систем. Там же производится и авторизация.

С чего начать?

- Если у вас есть лицензионный ключ, откройте страницу **Персональные данные** и зарегистрируйте ваш ключ, используя форму на этой странице.
- После установки WEBO Site SpeedUp находится в **режиме отладки**. Если внешний вид сайта и логика его работы в порядке, включите **рабочий режим**. Больше информации о режиме отладки и рабочем режиме можно получить в разделе описания страницы Панель управления или страницы Состояние системы.
- Вы можете оставаясь в режиме отладки испробовать различные конфигурации, доступные на странице Настройка. Подберите ту конфигурацию, которая наиболее эффективна для вашего сайта.
- Если подходящей конфигурации не нашлось, вы можете создать собственную конфигурацию, чтобы в деталях настроить работу вашего сайта. Используйте советы в блоке Настройка на странице Панель управления, чтобы включить максимум возможностей по оптимизации в вашей конфигурации.
- Обратите внимание на сообщения, отображаемые в блоке Состояние сервера на странице Панель управления. Для достижения наилучшего результата необходимо избавиться ото всех выявленных проблем (если таковые вообще есть). Обратитесь к разделу Устранение неполадок и поддержка для более подробной информации о том, как решить наиболее часто возникающие проблемы.
- На странице Кэш можно получить подробную информацию о содержимом кэша. И не забывайте обновлять кэш, если вы используете агрессивное клиентское или серверное кэширование.
- Получите еще большее ускорение, используя автоматизированные

инструменты для оптимизации изображений и статического сжатия (они расположены в блоке Инструменты оптимизации на странице Панель управления).

- Узнайте об обновлениях приложения на вкладке Обновления, на странице Состояние системы. При помощи этой страницы вы можете устанавливать даже ранние бета-версии WEBO Site SpeedUp.
- Будьте в курсе последних новостей из нашего официального блога, используя блок Новости на Панели управления, или подпишитесь при помощи [RSS](#). Так вы сможете узнать о новых технических статьях, масштабных обновлениях приложения, специальных предложениях и акциях и многом другом.
- Вы всегда можете отправить нам отзывы или предложения со страницы О WEBO Site SpeedUp — мы заинтересованы в повышении эффективности, качества и удобства использования WEBO Site SpeedUp.

Пожалуйста, не забудьте, что только в Полной версии WEBO Site SpeedUp ускорение вашего сайта может быть максимальным. Взгляните на страницу сравнения версий.

Детальное описание интерфейса

- Панель управления (с подстраницами инструментов для статического сжатия и оптимизации изображений, доступными только в Полной версии).
- Настройка
- Состояние системы (со вкладками: Состояние, Параметры, Обновления, Установка и Удаление)
- Персональные данные

3.1. Панель управления

Панель управления является страницей по умолчанию в панели администратора WEBO Site SpeedUp. На этой странице находится определенное количество настраиваемых блоков, отражающих наиболее значимую информацию о состоянии приложения и о работе сайта.

Для того, чтобы скрыть какой-либо блок, кликните по его закругленному краю. Чтобы восстановить блок, кликните по его имени в блоке Все блоки.

Ниже приведены короткие описания каждого блока.

Блоки на Панели Управления

Текущее состояние

Этот блок показывает текущий режим, в котором работает WEBO Site SpeedUp и позволяет перейти из одного режима в другой при помощи кнопок Включить и Отключить.

Существует два возможных режима: **режим отладки** и **рабочий режим**. В режиме отладки WEBO Site SpeedUp выполняет оптимизацию только в случае, если определенный GET-параметр передан вместе с запросом (этот параметр по умолчанию "web_optimizer_debug=1") или если установлен определенный cookie. В рабочем режиме WEBO Site SpeedUp выполняет оптимизацию при каждом запросе.

Блок аналогичен вкладке Состояние на странице Состояние системы.

Настройка

Этот блок отображает шкалу, показывающую насколько эффективна текущая конфигурация в рамках ограничений текущей версии WEBO Site SpeedUp (версии могут быть: Некоммерческая, Облегченная и Полная). Также этот блок дает несколько советов о том, как сделать сайт еще быстрее.

Состояние сервера

Этот блок сообщает обо всех проблемах и предупреждениях, которые влияют на работу WEBO Site SpeedUp. Чтобы узнать, как избавиться от подобных проблем обратитесь к разделу Устранение неполадок и поддержка.

Блок аналогичен вкладке Состояние на странице Состояние системы.

Состояние кэша

Этот блок показывает суммарный размер кэша WEBO Site SpeedUp и выводит краткие сведения о содержимом кэша. Кнопка Обновить позволяет полностью очистить и заново заполнить кэш (но только для главной страницы, используя цепочную оптимизацию).

Результаты оптимизации

Этот блок показывает насколько увеличилась скорость загрузки главной страницы сайта.

Инструменты оптимизации

Этот блок содержит ссылки на две подстраницы с инструментами оптимизации. Инструмент для **статического сжатия** позволяет создать в указанной директории .gz-версии всех файлов CSS и JS (а также некоторых других) для применения статического gzip-сжатия. Инструмент для **оптимизации изображений** позволяет уменьшить размер всех изображений в указанной директории вашего веб-сайта без потери качества. Эти инструменты доступны только в Полной версии WEBO Site SpeedUp и описаны более детально в конце данного раздела.

Быстрые ссылки

Этот блок содержит некоторое количество ссылок на наиболее полезные ресурсы, которые помогут сделать вашу работу с WEBO Site SpeedUp более эффективной.

Закажите настройку

Специалисты компании WEBO Software всегда готовы оказать квалифицированную помощь с установкой и настройкой WEBO Site SpeedUp. Если вы хотите быть уверенным в том, что приложение будет работать наиболее эффективно, отправьте нам запрос.

Обновления

Этот блок показывает информацию о последнем доступном стабильном обновлении WEBO Site SpeedUp и изменениях в нем. Когда такое обновление выходит вы можете кликнуть по кнопке Обновить и WEBO Site SpeedUp автоматически загрузит и установит обновление сохраняя текущую конфигурацию приложения.

Аналогично вкладке Обновления на странице Состояние системы.

Расскажите о нас

WEBO Site SpeedUp и ваш сайт сделал необычайно быстрым? Потратьте пару минут и расскажите об этом всему миру. Есть какие-то трудности с WEBO Site SpeedUp? Многим людям может быть интересен ваш опыт. Поделитесь своим опытом на ресурсах, приведенных в этом блоке.

Новости

Новости чаще всего касаются выхода технических статей на тему клиентской оптимизации, значительных обновлений приложения, акций и специальных предложений и многого другого. Оставайтесь на связи.

Инструмент для статического сжатия

Используя этот инструмент можно создать в указанной директории .gz-версии файлов CSS и JS (и некоторых других) для применения статического сжатия.

Во время сжатия время модификации (атрибут mtime) сжимаемых файлов устанавливается равным времени модификации первоначального (исходного) файла. Существующие файлы .gz обновляются тогда, когда время модификации исходного и существующего файлов различаются.

Для использования этого инструмента необходимо:

- Ввести абсолютный путь директории, содержащий файлы, которые необходимо сжать.
- Поставить или убрать флаг, указывающий, нужно ли приложению рекурсивно сжимать также все файлы в поддиректориях или нет.
- Кликнуть по кнопке Найти файлы.
- Просмотреть список найденных файлов и установить флаги для тех файлов, которые нужно сжать. Можно также использовать флаг в заголовке таблицы, чтобы установить флаг на всех файлах одновременно.
- Кликнуть по кнопке Сжать файлы, чтобы начать процесс сжатия.

По окончании процесса сжатия будет выведена таблица с результатами.

Необходимо заметить, что в случае, если некоторые директории были недоступны для записи, в них не будет создано сжатых .gz-версий исходных файлов.

Инструмент для оптимизации изображений

Используя этот инструмент можно уменьшить размер изображений без потери качества в указанной директории вашего сайта. Для каждого оптимизированного файла перед оптимизацией создается резервная копия. Файлы формата GIF заменяются файлами PNG в случае, когда последние могут быть меньше по размеру.

Для оптимизации может использоваться один из следующих сервисов: smush.it ([условия использования](#)) или [puppng](http://puppng.com) ([условия использования](#)).

Работа с этим инструментом осуществляется по тому же принципу, что и с инструментом для статического сжатия.

3.2. Настройка

Эта страница позволяет управлять конфигурациями WEBO Site SpeedUp. Каждая конфигурация представляет собой определенный набор параметров (параметров объединения, минимизации, сжатия, кэширования и т.д.), объединенных под одним именем.

Активная конфигурация — это та, которая в настоящее время применена и работает на вашем сайте. Она выделена ярким оранжевым фоном в списке конфигураций. Чтобы сделать конфигурацию активной нужно выбрать ее (кликнув по ней, вокруг нее появится оранжевая рамка) и после этого нажать кнопку *Сделать активной*.

Существует несколько предустановленных конфигураций: **Безопасная**, **Базовая**, **Оптимальная** и **Экстремальная**.

- **Безопасная** конфигурация, как это понятно из ее названия, представляет собой наиболее безопасный набор параметров WEBO Site SpeedUp для любого окружения. Она, однако, не практически не увеличивает производительность.
- **Базовая** является применяет основные и наиболее безопасные методы оптимизации. Проблемы с совместимостью этой конфигурации и окружением сайта возможны в единичных случаях.
- **Оптимальная** конфигурация наиболее сбалансирована. Достигается значительная производительность, однако это иногда может стать причиной изменений во внешнем виде или поведении страниц сайта. После применения этой конфигурации проверяйте внешний вид и работоспособность сайта.
- **Экстремальная** конфигурация обеспечивает наилучшую производительность. Большого можно добиться только при помощи тщательной ручной настройки. Но будьте внимательны, в зависимости от окружения эта конфигурация может значительно изменить внешний вид и поведение сайта. В связи с этим настоятельно рекомендуется проверять работоспособность этой конфигурации в режиме отладки, прежде чем переходить в рабочий режим.

Для того, чтобы получить больше сведений о конфигурации, выберите ее и вы увидите ее описание в поле, находящемся под ее названием. Для того, чтобы просмотреть все параметры какой-либо конфигурации, выберите ее и кликните по кнопке *Изменить*. Если вы хотите создать собственную конфигурацию с чистого листа, кликните по кнопке *Создать новую*, при этом за основу будет взята та конфигурация, вокруг которой находится оранжевая рамка. Чтобы удалить конфигурацию, выберите ее и кликните по иконке с изображением мусорной корзины.

Заметьте, что все предустановленные конфигурации не могут быть изменены или удалены и в случае их редактирования и сохранения будет создана новая конфигурация на их основе.

Все доступные параметры оптимизации приведены ниже. Заметьте, что некоторые из них доступны только в Облегченной или Полной версии WEBO Site SpeedUp. Больше подробностей можно получить на странице сравнения версий.

Объединение CSS

Объединение CSS-файлов

В зависимости от выбранной опции CSS либо не будет объединяться вовсе, либо будет объединяться только код, подключенный в тэге `<head>`, либо же будет объединяться весь CSS на странице. Весь объединенный код будет минимизироваться.

Объединять встроенный CSS-код

Будет объединяться весь CSS-код, подключаемый при помощи тэгов `<style>` и `<link>`. В противном случае будут объединяться только файлы, подключаемые при помощи тэгов `<link>`.

Объединять внешние CSS-файлы

Будут объединяться файлы, расположенные на любых хостах. В противном случае будут объединяться только файлы, расположенные на том же хосте, что и исходная страница.

Имя объединенного CSS-файла

Используйте эту опцию для того, чтобы на всех страницах сайта всегда вызывался один и тот же объединенный файл. Используйте только тогда, когда набор файлов CSS на страницах сайта одинаковый. Страницы с другим набором файлов CSS будут также загружать этот, ранее созданный файл. Файл с указанным именем статичен и после того, как он создан он изменяется лишь при ручном обновлении кэша.

Имя файла может содержать буквы латинского алфавита, цифры, знаки дефиса, подчеркивания и точки. Остальные символы будут автоматически удалены. Ко введенному имени файла может быть автоматически добавлено специальное окончание для форсирования сброса клиентского кэша в браузерах.

Хост CSS-файла(-ов)

Хост, с которого загружается объединенный CSS-файл. При включении соответствующей настройки в группе CDN с этого хоста будут загружаться все остальные CSS-файлы.

Исключать из объединения CSS-файлы

Указанные в этом поле файлы не будут включаться в объединенный файл. Необходимо указывать только имена файлов, а не абсолютные пути к ним.

Включать в объединенные CSS-файлы дополнительный код

Код введенный в это поле будет добавлен в конце каждого объединенного файла. Это поле позволяет задать дополнительные стили для страниц сайта, на котором работает WEBO Site SpeedUp.

Объединение JavaScript

Объединение JavaScript-файлов в тэгах <head>

В зависимости от выбранной опции JavaScript либо не будет объединяться вовсе, либо будут объединяться JavaScript-файлы, подключенные в тэге <head>.

Объединение JavaScript-файлов в тэгах <body>

В зависимости от выбранной опции JavaScript либо не будет объединяться вовсе, либо будут объединяться JavaScript-файлы, подключенные в тэге <body>.

Объединять встроенный JavaScript-код в тэгах <head>

Будет объединяться встроенный JavaScript-код в тэге <head>. В противном случае, будет объединяться только код, подключаемый при помощи <script src="...">.

Объединять встроенный JavaScript-код в тэгах <body>

Будет объединяться встроенный JavaScript-код в тэге <body>. В противном случае, будет объединяться только код, подключаемый при помощи <script src="...">.

Объединять внешние JavaScript-файлы

Будут объединяться файлы, расположенные на любых хостах. В противном случае будут объединяться только файлы, расположенные на том же хосте, что и исходная страница.

Имя объединенного JavaScript-файла

Используйте эту опцию для того, чтобы на всех страницах сайта всегда вызывался один и тот же объединенный файл. Используйте только тогда, когда набор файлов JavaScript на страницах сайта одинаковый. Страницы с другим набором файлов JavaScript будут также загружать этот, ранее созданный файл. Файл с указанным именем статичен и после того, как он создан он изменяется лишь при ручном обновлении кэша.

Имя файла может содержать буквы латинского алфавита, цифры, знаки дефиса, подчеркивания и точки. Остальные символы будут автоматически удалены. Ко введенному имени файла может быть автоматически добавлено

специальное окончание для форсирования сброса клиентского кэша в браузерах.

Хост JavaScript-файла(-ов)

Хост, с которого загружается объединенный JavaScript-файл. При включении соответствующей настройки в группе CDN с этого хоста будут загружаться все остальные JavaScript-файлы.

Исключать из объединения JavaScript-файлы

Указанные в этом поле файлы не будут включаться в объединенный файл. Необходимо указывать только имена файлов, а не абсолютные пути к ним.

Переносить вызов объединенного JavaScript-файла перед `</head>`

Вызов объединенного файла JavaScript будет переноситься к закрывающему тэгу `</head>`.

Безопасный режим объединения

При этом режиме все вызовы внешних файлов будут обернуты в конструкции try-catch (с подключением отдельных файлов, если их содержимое не удастся выполнить в общем контексте). Это снижает JavaScript-производительность, но гарантирует работоспособность всех используемых библиотек.

Удалять дубликаты

Все замеченные дубликаты известных библиотек (jQuery, Prototype, MooTools) будут удаляться из объединения. Это уменьшает размер итогового файла и сокращает время инициализации JavaScript, но в некоторых случаях может нарушить целостность клиентской логики.

Маска объединения скриптов

Вы можете задать маску для включения или исключения тегов script из итогового объединенного файла. Например, у вас идут 3 тега script, которые вы хотите включить в итоговый файл, а за ними — два, которые вы не хотите включать. В этом случае маска будет xxx00. Если не задано никакой маски, соответствующей текущему тегу script в документе, он будет включаться или исключаться согласно остальным правилам.

Минимизация

Минимизация CSS-файлов

Из всех объединенных файлов CSS будут удаляться лишние пробелы, символы табуляции, переносы строк и комментарии.

Минимизировать JavaScript-файлы

Из всех объединенных файлов JavaScript будут удаляться лишние пробелы, символы табуляции, переносы строк и комментарии. От выбора библиотеки зависит алгоритм минимизации, а следовательно финальный размер минимизированных файлов. В зависимости от особенностей минимизируемого кода более эффективный результат может быть достигнут с той или иной библиотекой.

Исключить из минимизации следующие файлы

При объединении JavaScript-кода указанные в этом поле файлы не будут минимизированы (но будут объединены согласно остальным настройкам группы «Объединение JavaScript»).

Минимизировать HTML

Из кода страниц будут удаляться двойные пробелы, двойные переводы строк, пустые символы в начале каждой строки и пробелы перед окончанием тэга. Все операции будут проводиться за исключением содержимого тэгов `<pre>`, `<textarea>`, `<script>`.

Сжать HTML до 1 строки

Код итоговой страницы будет вытянут в одну строку. С одной стороны подобная мера позволяет получить дополнительный выигрыш за счет удаления избыточных пробелов, табуляций и переводов строк, но с другой стороны требует значительных серверных ресурсов и должна быть использована с осторожностью.

Удалить HTML-комментарии

Все HTML-комментарии будут удаляться, что может повлечь за собой неработоспособность условных комментариев, а также ряда счетчиков.

Gzip-архивирование

Применить gzip для CSS

Все CSS-файлы будут передаваться в сжатом виде.

Применить gzip для JavaScript

Все JavaScript-файлы будут передаваться в сжатом виде.

Применить gzip для шрифтов

Все файлы шрифтов (.eot, .ttf, .otf и др.) будут передаваться в сжатом виде.

Применить `gzip` для HTML

Все HTML-файлы будут передаваться в сжатом виде.

Использовать `zlib`

Для `gzip`-сжатия будет использована PHP-библиотека `zlib`.

Проверять возможность `gzip` через `cookie`

WEBOSiteSpeedUp будет осуществлять дополнительную проверку на поддержку `gzip` в браузере, и если она будет определена, данные будут передаваться в сжатом виде вне зависимости от заголовка `Accept-Encoding`.

Применять `deflate` вместо `gzip` для IE6, IE7

В ряде ситуаций использование `gzip` в браузерах IE6 и IE7 может привести к некорректному отображению страницы. Данная опция позволяет использовать для этих браузеров альтернативный алгоритм сжатия.

Клиентское кэширование

Кэшировать CSS-файлы на клиенте

Всем CSS-файлам будут присваиваться кэширующие заголовки, установленные на далекое будущее.

Кэшировать JavaScript-файлы на клиенте

Если эта опция включена, всем JavaScript-файлам будут присваиваться кэширующие заголовки, установленные на далекое будущее.

Кэшировать изображения на клиенте

Всем изображениям будут присваиваться кэширующие заголовки, установленные на далекое будущее

Кэшировать файлы шрифтов на клиенте

Всем файлам шрифтов будут присваиваться кэширующие заголовки, установленные на далекое будущее.

Кэшировать видео-файлы на клиенте

Всем видео-файлам будут присваиваться кэширующие заголовки, установленные на далекое будущее. Эта опция может быть применена только при помощи `.htaccess`.

Кэшировать остальные файлы на клиенте

Всем остальным файлам будут присваиваться кэширующие заголовки,

установленные на далекое будущее. Эта опция может быть применена только при помощи `.htaccess`.

Кэшировать HTML на клиенте

Всем HTML-файлам будут присваиваться кэширующие заголовки, установленные в значение, заданное в поле «Время существования клиентского кэша для HTML-файлов».

Время существования клиентского кэша для HTML-файлов

Время, на которое кэшируются файлы HTML. Отсутствие значения означает нулевое время кэширования.

Кэшировать внешние файлы

Внешние файлы, подключаемые на страницах, будут отдаваться с того же хоста, что и запрашиваемая страница с заголовком `Expires` при помощи прокси-скрипта `/cache/wo.static.php` из директории `Site SpeedUp`. Для примера, вызов внешнего скрипта, который может находиться на странице будет преобразован из такого:

```
<script type="text/javascript" src="http://external-host.com/test.js"></script>
```

вот в такой:

```
<script type="text/javascript" src="/wordpress/wp-content/plugins/web-optimizer/cache/wo.static.php?http://external-host.com/test.js"></script>
```

.htaccess

Использовать `.htaccess`

Создает в корневой директории сайта файл `.htaccess`, либо модифицирует существующий файл, сохраняя все его исходное содержимое, а также создавая резервную копию исходного файла. Содержимое файла `.htaccess` изменяется в зависимости от других опций.

Расположить `.htaccess` в локальной (не корневой) директории

Файл `.htaccess` будет расположен в локальной папке сайта, а не корневой директории хоста.

Использовать `mod_deflate` + `mod_filter`

Требуется для динамического сжатия файлов и является альтернативой использованию `mod_gzip`.

Использовать `mod_gzip`

Требуется для динамического сжатия файлов и является альтернативой `mod_deflate` + `mod_filter`.

Использовать `mod_expires`

Требуется для выставления заголовков, отвечающих за клиентское кэширование.

Использовать `mod_headers`

Требуется для обеспечения корректной обработки сжатых файлов на прокси-серверах и в старых браузерах.

Использовать `mod_setenvif`

Требуется для обеспечения корректной обработки сжатых файлов на прокси-серверах и в старых браузерах.

Использовать `mod_rewrite`

Требуется для статического сжатия файлов или форсированного кэширования.

Использовать `mod_mime`

Требуется для статического сжатия файлов.

Обратная ссылка

Добавить ссылку на WEBO Site SpeedUp

Ссылка на WEBO Site SpeedUp является обязательным требованием для некоммерческой версии и может быть убрана только в платной версии.

Изображение обратной ссылки

Имя файла с логотипом WEBO Site SpeedUp. Доступные файлы находятся в следующей директории: <путь к WEBO Site SpeedUp>/web-optimizer/images/. Если это поле оставлено пустым, вместо изображения будет отображаться текст, указанный в поле «Текст обратной ссылки».

Текст обратной ссылки

Если поле «Изображение обратной ссылки» заполнено, текст этого поля будет использоваться в качестве альтернативного текста для изображения. В противном случае это будет непосредственно текст ссылки.

Стили для размещения ссылки

Эти стили будут присвоены ссылке на WEBO Site SpeedUp. Вы можете вручную

задать положение ссылки, ее цвет, фон, размер и т.д.

Добавлять `<!--wss-->` в `body` оптимизированных страниц

Наличие `<!--wss-->` сигнализирует о том, что WEBO Site SpeedUp успешно обработал текущую страницу, и используется во внутренних алгоритмах.

Добавлять счетчик времени загрузки

При сборе статистики посещаемости по сайту через Google Analytics данные о времени загрузки сайта будут попадать в События.

A/B тестирование

Заданный процент пользователей сайта получит неоптимизированные страницы. Все данные о результатах будут отправлены в Google Analytics.

Производительность

Не проверять время изменения файлов

Будет получен дополнительный выигрыш в быстродействии сервера, однако для обновления закешированных файлов у клиента будет необходимо либо изменить вызовы этих файлов (их URL), либо вручную обновить кэш.

Не проверять наличие файлов

Если эта опция включена, проверка на существование файлов в кэше производиться не будет, а версия всех файлов будет определяться значением в поле «Версия кэша». В этом случае для обновления закешированных файлов у клиента необходимо изменить версию кэша. Если эта опция отключена, проверка будет производиться.

Версия кэша

Версия кэша определяет версию всех файлов кэша. Для обновления закешированных файлов у клиента необходимо изменить значение этого поля.

Не применять регулярные выражения

Регулярные выражения отрицательно влияют на быстродействие сервера и могут быть заменены более простыми проверками, однако в этом случае вероятность некорректного разбора документов, не соответствующих стандартам, будет больше.

Использовать файлы в кэше для всех браузеров

Все браузеры будут получать одинаковый CSS-, JavaScript- и HTML-код, благодаря чему будет возможно безопасное использование внешних кэширующих механизмов, однако при этом станет невозможным использование

ряда технологий оптимизации, таких как data:URI.

Восстанавливать CSS-свойства

Все недостающие CSS-свойства, которые могут помочь более эффективно создавать CSS Sprites и data:URI будут определяться при помощи анализа всего CSS-кода. Это позволит уменьшить размер кэша, но может привести к существенным процессорным издержкам в случае большого количества CSS-правил.

Срок хранения файлов в кэше

Вы можете ограничить общий размер кэша при помощи срока хранения файлов в нем в днях. Нулевое значение отключает данное ограничение.

Отделить HTTPS-кэш от HTTP

Все HTML-файлы, запрашиваемые по HTTPS (SSL-соединение), будут храниться отдельно от обычных файлов, запрошенных по HTTP. Это увеличивает размер кэша, но гарантирует целостность сайта при запросах с использованием различных соединений.

Кэширующий движок

Вы можете выбрать один из доступных на данном сервере движков для кэширования HTML-файлов. Все движки кроме файловой системы хранят данные в оперативной памяти, что уменьшает время доступа к ним.

Масштабировать HTML-изображения

Все HTML-изображения на страницах сайта будут приведены к фактически используемым размерам, это может значительно сократить размер данных при загрузке сайта.

data:URI

Использовать data:URI

Фоновые изображения будут преобразовываться в формат base64 и подставляться в CSS-файлы для всех браузеров, поддерживающих технологию data:URI.

Использовать mhtml

Фоновые изображения будут преобразовываться в формат mhtml и подставляться в CSS-файлы для всех версий Internet Explorer, не поддерживающих технологию data:URI.

Максимальный размер `data:URI`

Изображения, размер которых превышает значение, заданное в этом поле, не будут преобразованы в формат `base64`. Отсутствие значения или нулевое значение означает отсутствие ограничения.

Максимальный размер `mhtml`

Изображения, размер которых превышает значение, заданное в этом поле, не будут преобразованы в формат `mhtml`. Отсутствие значения или нулевое значение означает отсутствие ограничения.

Исключить из `data:URI` файлы

Изображения, указанные в этом поле, не будут преобразованы в формат `base64`. Пожалуйста, указывайте только имена файлов, а не полные пути к ним.

Исключить из `mhtml` файлы

Изображения, указанные в этом поле, не будут преобразованы в формат `mhtml`. Пожалуйста, указывайте только имена файлов, а не полные пути к ним.

Отделить картинки от CSS-кода

Преобразованный код CSS и изображения в форматах `base64` и `mhtml` будут храниться в отдельных файлах. Это должно улучшить их кэширующее поведение.

CSS Sprites

Применять CSS Sprites

Фоновые изображения будут объединяться по технологии CSS Sprites, а CSS-код, отвечающий за их вывод, будет соответствующим образом преобразовываться.

«Агрессивный» режим создания CSS Sprites

Число изображений и размер изображений, созданных при помощи технологии CSS Sprites будет меньше, однако возможно появление артефактов на страницах.

Не применять CSS Sprites для IE6

IE6 будет получать собственный файл CSS без применения технологии CSS Sprites.

Максимальная ширина и высота изображений

Изображения, ширина или высота которых превышает значение, заданное в этом поле, не будут преобразованы по технологии CSS Sprites. Отсутствие

значения или нулевое значение означает отсутствие ограничения.

Максимальная ширина и высота HTML-изображений

Sprites будут ограничены заданным значением. Отсутствие значения или нулевое значение означает отсутствие ограничения.

Объединять HTML-изображения

Большое количество небольших HTML-изображений может быть также объединено в один файл для уменьшения количества запросов. При этом в HTML-документе вместо исходных изображений появляется прозрачное (по возможности вставленное через data:URI), фоном которого является исходное HTML-изображение.

Объединять изображения только для текущей страницы

HTML-изображения могут объединяться только для текущей страницы (это уменьшит размер каждого объединенного файла, но увеличит их количество), либо для всех страниц сайта. В последнем случае на всем сайте будет использоваться только 1 изображение, но его размер может быть значительным.

Максимальная ширина и высота HTML-изображений

HTML-изображения, ширина или высота которых превышает значение, заданное в этом поле, не будут преобразованы по технологии CSS Sprites. Отсутствие значения или нулевое значение означает отсутствие ограничения.

Включение / исключение файлов для CSS Sprites

Изображения, указанные в поле ниже, либо не будут преобразованы по технологии CSS Sprites, либо будут преобразованы только они.

Добавить свободное место в CSS Sprites

Некоторые изображения в файлах, созданных по технологии CSS Sprites, будут окружаться пустым пространством, чтобы избежать появления артефактов при масштабировании страницы пользователем. Размер файлов изображений при этом несколько увеличится.

Формат изображений

Если выбрана опция автоматического определения формата, риск появления артефактов в изображениях будет минимальным. Если предпочтение отдается формату JPEG, соотношение размер/качество для полноцветных изображений будет наилучшим, однако использование прозрачности будет невозможным.

Серверное кэширование

Применять серверное кэширование

Страницы будут кэшироваться на сервере на время, заданное в поле «Время существования серверного кэша для HTML-файлов». Применение этой опции позволяет во много раз ускорить загрузку страниц, долго генерирующихся на сервере, однако это целесообразно только для статических страниц без динамически изменяемых параметров.

Время существования HTML-файлов серверного кэша

По истечению этого времени закэшированные на сервере страницы будут создаваться заново.

Время кэширования корзины интернет-магазина

В течение этого времени все данные о пользовательской корзине будут храниться локально у пользователя (в cookie или localStorage).

Применять ранний сброс части страницы

При наличии в кэше на сервере запрошенной страницы, некоторое число начальных байтов этой страницы (заданное в поле «Размер сбрасываемой части страницы») будет отправляться и «сбрасываться» клиенту раньше остального содержимого страницы. Благодаря этой опции браузер может раньше получить первую часть страницы и раньше начать загрузку требуемых ресурсов.

Применять ранний сброс фиксированного числа байт

Размер рано сбрасываемой части страницы может быть фиксированным для избежания проблем с браузерами, имеющими ограничения на минимальный или максимальный размер получаемой части страницы. Отсутствие значения или нулевое значение приведет к раннему сбросу содержимого страницы от ее начала до закрывающего тэга `</head>`.

Исключать из HTML-кэша URL

Часто серверное кэширование неприменимо для страниц с динамически изменяемыми параметрами, например, для страниц аккаунта пользователя, страниц статистики и других. Данное поле позволяет указать части (маски) URL страниц, которые не следует кэшировать на сервере.

Применять серверное кэширование для указанных USER AGENTS

Данное поле позволяет указать список USER AGENTS, которым будут отправляться строго закэшированные файлы. Например, за счет отдачи закэшированных файлов всем поисковым роботам может быть снижена нагрузка на сервер.

Список COOKIE для исключения при кэшировании

Вы также можете отключить кэширование страниц для пользователей, у которых заданы указанные COOKIE. Это бывает полезно для авторизованных пользователей или при работе с покупательской корзиной.

Список GET-параметров для исключения при кэшировании (через пробел)

Вы можете указать список дополнительных GET-параметров, не влияющих на содержание страницы (например, идентификаторы рекламных кампаний). Их исключение из составления ключа для кэширования позволит уменьшить размер кэша и повысить его эффективность.

Экстремальный режим

В экстремальном режиме все HTML-документы отдаются напрямую из кэша, в обход системы управления сайтами. Это существенно увеличивает производительность сайта (что особенно необходимо при большой посещаемости), но обновление кэша при этом может производиться только вручную. По умолчанию экстремальный режим кэширования работает только на системах с веб-серверами, поддерживающими технологию .htaccess (Apache, LiteSpeed). Для поддержки других серверов нужно вручную изменить их конфигурацию, согласно разделу Интеграция с веб-сайтом.

Удалять старые файлы

После заданного времени существования серверного кэша все записи в нем будут удалены. Это повышает эффективность кэширования и позволяет форсировать обновление сайта для пользователей.

Кэширование БД

Кэшировать запросы к базе данных

Все запросы, выполнение которых занимает больше указанного времени, будут закэшированы, чтобы ускорить отображение страниц сайта для всех пользователей. Для корректной работы кэширования SQL-запросов необходима установка WEBO Site SpeedUp на уровне системного расширения (а не в качестве отдельного приложения).

Время выполнения тяжелых запросов (мс)

Если запросы выполняются больше указанного времени (в мс), то результат их выполнения будет закэширован.

Срок хранения кэша (с)

По истечению этого времени все запросы будут заново запрашиваться у текущей базы данных.

Исключить таблицы (через пробел)

Исключить таблицы (через пробел)

Вы можете исключить ряд таблиц из кэширования запросов. Для этого необходимо перечислить их названия (без префикса) через пробел. SQL-запросы к таким таблицам не будут кэшироваться.

Ненавязчивый JavaScript

«Ненавязчивый» JavaScript

В базовом режиме все внешние JavaScript-блоки на сайте будут загружаться по событию `onDOMloaded`. Если же включен улучшенный режим, то загрузка будет производиться по событию `window.onload` в соответствии с остальными опциями данного раздела.

Доступные настройки: базовый режим и улучшенный режим.

Переносить вызов объединенного JavaScript-файла перед `</body>`

Вызов объединенного файла будет перенесен к закрывающему тэгу `</body>`. Эта опция имеет больший приоритет, чем «Переносить вызов объединенного JavaScript-файла перед `</head>`».

Переместить весь JavaScript-код перед `</body>`

Весь JavaScript-код будет переноситься к закрывающему тэгу `</body>` в том же порядке, в котором этот код встречался на исходной странице.

Перенести вызовы JavaScript-информеров перед `</body>`

Код всех JavaScript-информеров будет переноситься к закрывающему тэгу `</body>`.

Перенести вызовы JavaScript-счетчиков перед `</body>`

Код всех JavaScript-счетчиков будет переноситься к закрывающему тэгу `</body>`.

Перенести рекламные вызовы JavaScript перед `</body>`

Код всех рекламных блоков (контекстной и баннерной рекламы) будет переноситься к закрывающему тэгу `</body>`.

Отложить загрузку фреймов

Код всех фреймов будет переноситься к закрывающему тэгу `</body>`.

Загружать фоновые изображения по событию `DOMready`

Загрузка фоновых изображений будет откладываться до события `DOMReady`. Это увеличит скорость появления страницы на экране браузера.

Предзагружать CSS- или JavaScript-файлы

Все указанные URL будут загружены по событию `window.onload`, чтобы ускорить просмотр следующих страниц сайта.

Предзагружать страницы

Все ресурсы на указанных URL будут загружены по событию `window.onload` во фреймах, чтобы ускорить просмотр следующих страниц сайта.

CDN

Распределять изображения

Все файлы изображений, вызываемые на страницах будут автоматически распределяться по нескольким хостам. Так, например, вызовы файлов `http://site.ru/i/logo.png` и `/i/bg.jpg` могут быть заменены вызовами `http://i1.site.ru/i/logo.png` и `http://i2.site.ru/i/bg.jpg`, при условии, что хосты `i1` и `i2` доступны и указаны в поле «Использовать хосты».

Автоматически определять доступные хосты

Доступные хосты будут определяться автоматически (по существованию на них картинок сайта).

Использовать хосты

Для распределения изображений будут использоваться хосты, указанные в этом поле. Хостов может быть не более четырех.

Распределять CSS-файлы

Все CSS-файлы будут выдаваться с хоста, определенного как «Хост CSS-файлов» в группе настроек «Объединение CSS».

Распределять JavaScript-файлы

Все CSS-файлы будут выдаваться с хоста, определенного как «Хост JavaScript-файлов» в группе настроек «Объединение JavaScript».

Исключить следующие файлы из распределения (через пробел)

Вы можете указать список файлов (например, динамических) для исключения при распределении изображений по нескольким хостам.

Использование CDN

Вы можете самостоятельно настроить использование CDN (при помощи настроек данной группы и хостов для CSS- и JavaScript-файлов) или выбрать один из имеющихся вариантов.

Доступные настройки: самостоятельная настройка, текущая CDN (cdn.website.ru), Coral CDN (.nyud.net).

FTP-доступ для загрузки файлов

В случае использования платных CDN (например, EdgeCast) необходимо указать строку FTP-доступа для загрузки новых файлов вида user:password@host (или user:password@host:port). Однако, только объединенные CSS и JS файлы могут быть загружены по FTP, поскольку иначе в виду медленной работы FTP быстродействие будет крайне низким.

Хост, доступный по HTTPS:

Если ваш сайт доступен по протоколу HTTPS, то для корректного использования CDN необходим хост, имеющий SSL-сертификат. Этот хост будет использоваться для распределения всех типов файлов при защищенных (HTTPS) запросах.

3.3. Состояние системы

Эта страница состоит из четырех вкладок:

Состояние

На этой вкладке выводится текущий режим WEBO Site SpeedUp. Есть также возможность переключить режим нажатием кнопки Включить или Отключить.

Существует два возможных режима: **режим отладки** и **рабочий режим**. В режиме отладки WEBO Site SpeedUp выполняет оптимизацию только в случае, если определенный GET-параметр передан вместе с запросом (этот параметр по умолчанию "web_optimizer_debug=1") или если установлен определенный cookie. В рабочем режиме WEBO Site SpeedUp выполняет оптимизацию при каждом запросе.

На этой вкладке также выводятся предупреждения и сообщения об ошибках системы, которые могут повлиять на производительность WEBO Site SpeedUp. Узнать, как можно избавиться от таких проблем можно в разделе Устранение неполадок и поддержка.

Предупреждения, относящиеся к модулям Apache modules и ограничениям памяти не относятся к WEBO Site SpeedUp и являются следствием текущей конфигурации сервера. Свяжитесь с поставщиком услуг хостинга для того, чтобы устранить эти предупреждения. За счет тщательно проработанного алгоритма «плавной деградации» WEBO Site SpeedUp способен работать практически на любом серверном окружении, используя максимум доступных путей для оптимизации.

Эта вкладка аналогична блокам **Текущее состояние** и **Состояние сервера** на странице Панель управления.

Параметры

Эта вкладка позволяет просмотреть и изменить системные параметры самого приложения WEBO Site SpeedUp. В большинстве случаев эти параметры выставляются автоматически и не должны изменяться. Изменяйте их только если вы знаете, зачем вы это делаете. Сами параметры описаны далее.

Кэш

Эта вкладка содержит информацию о содержимом кэша. Файлы кэша — это оптимизированные файлы, которые WEBO Site SpeedUp получает от вашей системы и передает конечным пользователям, делая тем самым процесс оптимизации гораздо более быстрым.

Кнопка Обновить кэш позволяет полностью очистить содержимое Кэша и заново заполнить его содержимым, так, будто вы открыли главную страницу вашего

сайта.

Обновление кэша обычно требуется:

- после перенастройки WEBO Site SpeedUp,
- после изменения содержимого сайта, в случае использования функции серверного кэширования HTML.

Аналогично блоку **Кэш** на странице Панель управления.

Адрес сайта

Доменное имя или IP-адрес оптимизируемого сайта. Пример: `mysite.ru`. Доменное имя используется для определения внешних ресурсов, а также как хост для объединенных CSS- или JS-файлов. Вы также можете использовать (например, для статических зеркал для указанных файлов) следующие хосты.

- `js.mysite.ru`
- `css.mysite.ru`
- `beta.mysite.ru`
- `test.mysite.ru`
- `local.mysite.ru`
- `stat.mysite.ru`
- `stat1.mysite.ru`
- `static.mysite.ru`
- `dev.mysite.ru`
- `www2.mysite.ru`
- `cdn.mysite.ru`

Путь к корневой директории сайта

Абсолютный путь к корневой директории вашего сайта.

Путь к корневой директории хоста (`DOCUMENT_ROOT`)

Absolute path to the root directory of the website's host.

Путь к директории CSS-кэша

В этой директории располагаются файлы CSS-кэша.

Путь к директории JavaScript-кэша

В этой директории располагаются файлы JavaScript-кэша.

Путь к директории HTML-кэша

В этой директории располагаются файлы HTML-кэша.

Защитить установку WEBO Site SpeedUp с помощью `htpasswd`

Обеспечивает дополнительную безопасность WEBO Site SpeedUp при помощи механизма, основанного на применении технологии HTTP Basic Authorization и

файлов .htaccess и .htpasswd.

Логин для защиты установки WEBO Site SpeedUp с помощью .htpasswd

Для защиты с помощью .htpasswd требуются логин и пароль. В качестве логина используется значение этого поля. В качестве пароля автоматически используется пароль от установки WEBO Site SpeedUp.

Логин пароль для доступа по HTTP Basic Authorization

В случае, если доступ на сайт, где используется WEBO Site SpeedUp, ограничен посредством HTTP Basic Authorization, необходимо указать логин и пароль, чтобы WEBO Site SpeedUp мог обрабатывать необходимые ресурсы с сайта.

Исключить обработку следующих URL

В некоторых случаях необходимо исключить часть разделов сайта из логики работы WEBO Site SpeedUp. В этом случае необходимо задать характерные части этих разделов (маски) через пробел.

Обновления

Этот блок показывает информацию о последнем доступном стабильном обновлении WEBO Site SpeedUp и изменениях в нем. Когда такое обновление выходит вы можете кликнуть по кнопке Обновить и WEBO Site SpeedUp автоматически загрузит и установит обновление сохраняя текущую конфигурацию приложения.

Также отсюда можно установить последнюю бета-версию приложения. Будьте осторожны с этим, но помните, что вы можете снова восстановить стабильную версию приложения, если что-то пойдет не так.

Эта вкладка аналогична блоку Обновления на странице Панель управления.

Установка и Удаление

Эта вкладка активна только если WEBO Site SpeedUp установлен в качестве отдельного приложения.

Если WEBO Site SpeedUp установлен на систему, входящую в список поддерживаемых систем, на этой вкладке будут сведения об изменениях, происходящих во время установки WEBO Site SpeedUp в исходных файлах системы.

Также тут есть две кнопки, позволяющие Переустановить или Удалить все внесенные изменения. Когда все внесенные изменения Удалены, система восстанавливается до исходного состояния (заметьте, что все файлы WEBO Site SpeedUp, включая файлы конфигурации и кэша будут сохранены), но вы также всегда можете снова применить все изменения нажав кнопку Установить.

Если WEBO Site SpeedUp установлен на систему, не входящую в список поддерживаемых систем, на этой вкладке будут находиться инструкции, касающиеся ручной установки и удаления WEBO Site SpeedUp.

3.4 Персональные данные

Эта страница содержит единственную форму, все поля которой детально описаны ниже.

Поля страницы Персональные данные

Лицензионный ключ

Это ключ, который делает доступными все функции Облегченной и Полной версии (в зависимости от вашей лицензии). Лицензии на Облегченную и Полную версии WEBO Site SpeedUp предоставляются на неограниченный период (сравнение версий). Для Некоммерческой версии WEBO Site SpeedUp данное поле заполнять не требуется.

Регистрация лицензионного ключа осуществляется автоматически. Достаточно ввести правильный ключ в соответствующее поле и нажать кнопку Сохранить. Вы можете задавать любые вопросы, касающиеся лицензий WEBO Site SpeedUp, используя [контактные данные](#), приведенные на официальном сайте.

Для всех плагинов кроме Joomla! новые функции будут видны и доступны сразу после сохранения и регистрации лицензионного ключа. Для плагина Joomla! (из-за особенностей его API) все функции видимы всегда, но они становятся активными только после того, как будет введен и зарегистрирован правильный лицензионный ключ. Также для плагина Joomla! требуется обновить кэш WEBO Site SpeedUp.

E-mail

Этот адрес будет использован только для сообщения вам информации о критических обновлениях и специальных предложениях.

Разрешить использовать мои данные о результате оптимизации

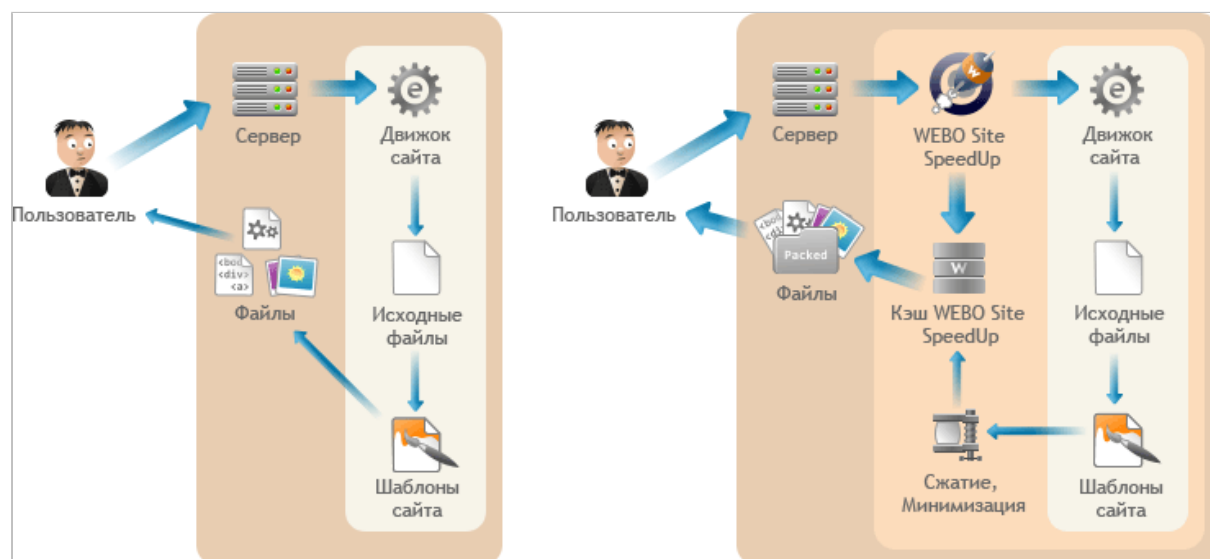
Статистическая информация об ускорении сайта может быть отправлена на серверы WEBO Software. Эта информация не будет опубликована и будет использоваться только для того, чтобы улучшить качество и эффективность WEBO Site SpeedUp. Никакие персональные данные не будут отправлены.

Если WEBO Site SpeedUp установлен в качестве отдельного приложения на этой странице также будут три поля, позволяющие сменить текущий пароль. В случаях, когда WEBO Site SpeedUp установлен в качестве плагина, проверка прав доступа производится естественным образом, на уровне той системы, в рамках которой работает плагин.

4. Принципы работы WEBO Site SpeedUp

Чтобы достичь наилучшего результата после оптимизации требуется хорошее понимание всех техник и принципов оптимизации.

WEBO Site SpeedUp работает как очень легкое проксирующее приложение для вашего PHP-сайта: он буферизует выходящий поток информации (или получает его при помощи API самой системы управления сайтом), применяет к нему все возможные оптимизационные мероприятия и выдает ускоренный HTML-код конечным пользователям. Все производимые действия проходят тщательную проверку качества и быстродействия, так что общая нагрузка на сервер не увеличивается. А если дополнительно использовать встроенные возможности серверного кэширования самого WEBO Site SpeedUp, то серверные издержки можно вообще свести к минимуму и увеличить запас прочности вашего веб-сервера. Ниже приведены схемы работы сайта без WEBO Site SpeedUp и с установленным WEBO Site SpeedUp соответственно.



В следующих разделах вы узнаете о том, каким образом WEBO Site SpeedUp интегрируется с любой CMS и даже сайтами из статичных HTML-документов, как он выполняет все действия по оптимизации и даже то, как вы можете расширить функциональность WEBO Site SpeedUp, дополнив его собственным расширением, API для расширений:

- Интеграция с веб-сайтом
- Алгоритм оптимизации
- CSS-правила для лучшего создания CSS Sprites
- API для расширений
- Системные файлы

4.1. Интеграция с веб-сайтом

Способы интеграции

WEBO Site SpeedUp может быть интегрирован на сайт при помощи нескольких подходов:

- Подход с использованием буфера
- Прямой подход
- Комплексный подход
- Подход при выключении

Подход с использованием буфера

Это наиболее стандартный путь использования WEBO Site SpeedUp. Самый первый вызов (с require общей библиотеки) заканчивается вызовом `ob_start()`. Затем все содержимое сайта, которое выводится (это может быть как простой вызов `echo $content`, так и сотни небольших `echo 'несколько тегов'`) будет буферизоваться до следующего вызова WEBO Site SpeedUp. Второй (и последний) вызов `$web_optimizer->finish()`; заканчивает буферизацию с `ob_get_clean()`. Затем весь контент (HTML-документ) анализируется и возвращается в оптимизированном виде. Обычно в коде это выглядит следующим образом:

```
require('.../web.optimizer.php');  
...  
echo ...  
...  
echo ...  
...  
$web_optimizer->finish();
```

Замечено несколько проблем, связанных с PHP-буферизацией различными подсистемами сайта, поэтому данный подход работает не всегда. Но обычно его можно использовать для любого сайта.

Прямой подход

В версии 0.5.9 появилась возможность анализировать HTML-содержимое без какой-либо буферизации. Для этого необходимо добавить следующие строки кода:

```
$not_buffered = 1;  
require('.../web.optimizer.php');  
$content = $web_optimizer->finish($content);
```

как мы видим, нужно передать WEBO Site SpeedUp содержимое для анализа, а затем WEBO Site SpeedUp вернет уже уже оптимизированным. Этот подход также можно использовать, если требуется интегрировать приложение в

процесс оптимизации по расписанию: здесь не происходит никакой буферизации.

Комплексный подход

Оба подхода могут быть использованы вместе. Хорошим примером этого является интеграция вызовов WEBO Site SpeedUp в плагин для WordPress:

```
/* основная функция для вызова на каждой странице */
function web_optimizer_init() {
    ob_start('web_optimizer_shutdown');
}
/* оборачивает выходной поток в процесс оптимизации */
function web_optimizer_shutdown ($content) {
    $not_buffered = 1;
    require(dirname(__FILE__) . '/web-optimizer/web.optimizer.php');
    return $web_optimizer->finish($content);
}
/* добавляем эту функции в соответствующий хук */
add_action('plugins_loaded', 'web_optimizer_init');
```

Подход с использованием буфера применен в функции `web_optimizer_init`, где буферизация инициализируется и откладывается до тех пор пока веб-страница не будет готова к отправке браузеру. В то же время функция `web_optimizer_shutdown` реализует прямой подход и возвращает оптимизированное содержимое после единственного вызова WEBO Site SpeedUp.

Подход при выключении

Есть еще один способ интегрировать WEBO Site SpeedUp с сайтом через регистрации функции выключения (через `ob_start`). Например (в самом начале файла `index.php`):

```
$not_buffered = 1;
require(dirname(__FILE__) . '/web-optimizer/web.optimizer.php');
function weboptimizer_shutdown ($content) {
    if (!empty($content)) {
        global $webo_request_uri;
        $_SERVER['REQUEST_URI'] = $webo_request_uri;
        $not_buffered = 1;
        require(dirname(__FILE__) . '/web-optimizer/web.optimizer.php');
        if (!empty($web_optimizer)) {
            $weboptimizer_content = $web_optimizer->finish($content);
        }
        if (!empty($weboptimizer_content)) {
            $content = $weboptimizer_content;
        }
    }
}
```



```
return $content;
}
}
ob_start('weboptimizer_shutdown');
```

Оптимизация по расписанию

Уже сейчас WEBO Site SpeedUp может быть встроен в схему публикации произвольного сайта в "статическом" режиме. Для этого необходимо открыть все страницы сайта с установленным WEBO Site SpeedUp, а потом просто скопировать выведенный HTML и кэширующие директории. Предположим, что WEBO Site SpeedUp установлен на `dev.site.ru`. Запустив, например, `wget` мы получим оптимизированный "слепок" сайта, который можно загрузить уже в рабочую систему.

```
wget -d -r -c http://dev.site.ru/
```

Изменения на сервере

Изменения в исходных файлах

В случае, если WEBO Site SpeedUp установлен в качестве отдельного приложения, в нескольких файлах, отвечающих за вывод HTML (в зависимости) автоматически добавляется две строки, в самом начале и в самом конце файла. Чаще всего так изменяется единственный файл `index.php` в корневой директории сайта.

В случае, если WEBO Site SpeedUp установлен в качестве плагина, чаще всего никакие системные файлы системы не изменяются кроме тех, которые всегда изменяются при установке любых плагинов в данной системе.

Единственный файл, который может быть изменен в любой системе это `.htaccess` в корневой директории сайта. Изменения производятся только в том случае, если опция **Использовать .htaccess** на вкладке `.htaccess tab` страницы Настройка включена.

WEBO Site SpeedUp автоматически делает резервные копии всех измененных файлов. Резервные копии отличаются от оригиналов дополнительным разрешением `.backup`. К примеру, имя файла `index.php` изменится на `index.php.backup` после установки WEBO Site SpeedUp, а новый файл `index.php` будет содержать необходимые для WEBO Site SpeedUp изменения.

WEBO Site SpeedUp никогда не изменяет какие-либо другие файлы, включая исходные коды системы, изображения, файлы CSS или JS без вашего участия. Оптимизированные файлы хранятся и отдаются из отдельных директорий кэша, таким образом, что оригинальные файлы всегда находятся в сохранности. Исключением является инструмент для Оптимизации изображений, но и он может быть запущен исключительно вручную.

Изменения в .htaccess

Если включена опция **Использовать .htaccess** на вкладке .htaccess на странице Настройка, WEBO Site SpeedUp автоматически изменит файл .htaccess для того, чтобы реализовать нужные действия по оптимизации. Все эти изменения описаны ниже.

mod_expires - все кэширующие заголовки

```
ExpiresActive On
```

Если требуется кэшировать HTML-файлы

```
<FilesMatch \.(html|xhtml+xml|shtml|phtml|php)$>
ExpiresDefault "access plus здесь_идет_таймаут_для_HTML seconds"
</FilesMatch>
ExpiresByType text/html A_HTML_таймаут
ExpiresByType text/xml A_HTML_таймаут
ExpiresByType application/xhtml+xml A_HTML_таймаут
ExpiresByType text/plain A_HTML_таймаут
```

Если требуется кэшировать CSS-файлы

```
<FilesMatch \.css$>
ExpiresDefault "access plus 10 years"
</FilesMatch>
ExpiresByType text/css A315360000
```

Если требуется кэшировать JavaScript-файлы

```
<FilesMatch \.js$>
ExpiresDefault "access plus 10 years"
</FilesMatch>
ExpiresByType text/javascript A315360000
ExpiresByType application/javascript A315360000
ExpiresByType application/x-javascript A315360000
ExpiresByType text/x-js A315360000
ExpiresByType text/ecmascript A315360000
ExpiresByType application/ecmascript A315360000
ExpiresByType text/vbscript A315360000
ExpiresByType text/fluffscript A315360000
```

Если требуется кэшировать изображения

```
<FilesMatch \.(bmp|png|gif|jpe?g|ico)$>
ExpiresDefault "access plus 10 years"
</FilesMatch>
ExpiresByType image/gif A315360000
ExpiresByType image/png A315360000
ExpiresByType image/jpeg A315360000
ExpiresByType image/x-icon A315360000
ExpiresByType image/bmp A315360000
```

Если требуется кэшировать шрифты

```
<FilesMatch \.(eot|ttf|otf|svg)$>
ExpiresDefault "access plus 10 years"
</FilesMatch>
ExpiresByType application/x-font-opentype A315360000
ExpiresByType application/x-font-truetype A315360000
ExpiresByType application/x-font-ttf A315360000
ExpiresByType application/x-font A315360000
ExpiresByType font/opentype A315360000
ExpiresByType font/otf A315360000
ExpiresByType application/vnd.oasis.opendocument.formula-template
A315360000
ExpiresByType image/svg+xml A315360000
ExpiresByType application/vnd.ms-fontobject A315360000
ExpiresByType font/woff A315360000
```

Если требуется кэшировать видео-файлы

```
<FilesMatch \.(flv|wmv|asf|asx|wma|wax|wmx|wm)$>
ExpiresDefault "access plus 10 years"
</FilesMatch>
ExpiresByType video/x-flv A315360000
ExpiresByType video/x-ms-wmv A315360000
ExpiresByType video/x-ms-asf A315360000
ExpiresByType video/x-ms-asx A315360000
ExpiresByType video/x-ms-wma A315360000
ExpiresByType video/x-ms-wax A315360000
ExpiresByType video/x-ms-wmx A315360000
ExpiresByType video/x-ms-wm A315360000
```

Если требуется кэшировать другие статические ресурсы

```
<FilesMatch \.(swf|pdf|doc|rtf|xls|ppt)$>
ExpiresDefault "access plus 10 years"
</FilesMatch>
ExpiresByType application/x-shockwave-flash A315360000
ExpiresByType application/pdf A315360000
ExpiresByType application/msword A315360000
ExpiresByType application/rtf A315360000
ExpiresByType application/vnd.ms-excel A315360000
ExpiresByType application/vnd.ms-powerpoint A315360000
```

mod_deflate + mod_filter (если отсутствует mod_gzip) - вся логика архивирования

```
AddOutputFilterByType DEFLATE text/html
AddOutputFilterByType DEFLATE text/xml
AddOutputFilterByType DEFLATE image/x-icon
```

gzip для CSS-файлов

```
AddOutputFilterByType DEFLATE text/css
```

gzip для JavaScript-файлов



```
AddOutputFilterByType DEFLATE text/javascript
AddOutputFilterByType DEFLATE application/javascript
AddOutputFilterByType DEFLATE application/x-javascript
AddOutputFilterByType DEFLATE text/x-js
AddOutputFilterByType DEFLATE text/ecmascript
AddOutputFilterByType DEFLATE application/ecmascript
AddOutputFilterByType DEFLATE text/vbscript
AddOutputFilterByType DEFLATE text/fluffscript
```

mod_gzip - вся логика gzip-сжатия, если mod_deflate отсутствует

Включаем модуль gzip

```
mod_gzip_on Yes
mod_gzip_can_negotiate Yes
mod_gzip_static_suffix .gz
AddEncoding gzip .gz
mod_gzip_update_static No
mod_gzip_keep_workfiles No
mod_gzip_minimum_file_size 500
mod_gzip_maximum_file_size 5000000
mod_gzip_maximum_inmem_size 60000
mod_gzip_min_http 1000
mod_gzip_handle_methods GET POST
mod_gzip_item_exclude reqheader "\"User-agent: Mozilla/4.0[678]\"
mod_gzip_dechunk No
```

Добавляем gzip для страницы

```
mod_gzip_item_include mime ^text/html$
mod_gzip_item_include mime ^text/plain$
mod_gzip_item_include mime ^image/x-icon$
mod_gzip_item_include mime ^httpd/unix-directory$
```

Добавляем gzip для CSS-файлов

```
mod_gzip_item_include mime ^text/css$
```

Добавляем gzip для JavaScript-файлов

```
mod_gzip_item_include mime ^text/javascript$
mod_gzip_item_include mime ^application/javascript$
mod_gzip_item_include mime ^application/x-javascript$
mod_gzip_item_include mime ^text/x-js$
mod_gzip_item_include mime ^text/ecmascript$
mod_gzip_item_include mime ^application/ecmascript$
mod_gzip_item_include mime ^text/vbscript$
mod_gzip_item_include mime ^text/fluffscript$
```

mod_headers - для предохранения прокси-серверов от gzip и
корректного выставления условного кэширования

Отменяем кэширование для архивов на прокси-серверах

```
<FilesMatch \.(css|js)$>
Header append Vary User-Agent
Header append Cache-Control private
</FilesMatch>
```

Отменяем заголовок Last-Modified (и добавляем вместо него ETag)

```
<FilesMatch \.(ico|pdf|flv|swf|jpe?g|png|gif|bmp|js|css)$>
Header unset Last-Modified
FileETag MTime
</FilesMatch>
```

mod_setenvif - для предохранения старых браузеров от gzip

Иключаем браузеры, которые некорректно обходятся с gzip

```
BrowserMatch ^Mozilla/4 gzip-only-text/html
BrowserMatch ^Mozilla/4\.0[678] no-gzip
BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
```

Определяем подходящий способ сжатия и тип браузера для применения экстремального сжатия

```
SetEnvIfNoCase accept-encoding deflate WSENC=.df
SetEnvIfNoCase accept-encoding gzip WSENC=.gz
BrowserMatch "MSIE 6" WSSBR=.ie6
BrowserMatch "MSIE 7" WSSBR=.ie7
BrowserMatch "MSIE 8" WSSBR=.ie8
BrowserMatch
"Android|BlackBerry|HTC|iPhone|iPod|LG|MOT|Mobile|NetFront|Nokia|Opera
Mini|Palm|PPC|SAMSUNG|Smartphone|SonyEricsson|Symbian|UP.Browser|webOS"
WSSBR=.ma
```

mod_rewrite + mod_mime (в дополнение к mod_deflate или mod_gzip)
-- для статического архивирования CSS- и JS- файлов

Добавляем кодировку для статических архивов

```
AddEncoding gzip .gz
```

Добавляем редиректы на физические файлы (имена файлов выставляются по времени их изменения, чтобы корректно сбрасывать кэш на клиенте)

```
RewriteRule ^(.*)\.wo[0-9]+\.(css|php)$ $1.$2
RewriteRule ^(.*)\.wo[0-9]+\.(js|php)$ $1.$2
```

Добавляем статическое архивирование для CSS-файлов

```
RewriteCond %{HTTP:Accept-encoding} gzip
RewriteCond %{HTTP_USER_AGENT} !Konqueror
RewriteCond %{REQUEST_FILENAME}.gz -f
RewriteRule ^(.*)\.css$ $1.css.gz [QSA,L]
```

Добавляем статическое архивирование для JavaScript-файлов

```

RewriteCond %{HTTP:Accept-encoding} gzip
RewriteCond %{HTTP_USER_AGENT} !Konqueror
RewriteCond %{REQUEST_FILENAME}.gz -f
RewriteRule ^(.*)\.js$ $1.js.gz [QSA,L]

```

Добавляем условия и редиректы для применения экстремального сжатия (на примере WordPress)

```

RewriteCond %{HTTP:Cookie} !^.*(comment_author_|wordpress|wp-
postpass_).*$
RewriteCond %{REQUEST_METHOD} !=POST
RewriteCond /path-to-document-root/wp-content/plugins/webo-site-
speedup/cache/%{HTTP_HOST}/
%{REQUEST_URI}%{QUERY_STRING}index%{ENV:WSSBR}.html%{ENV:WSSENC} -
f
RewriteRule (.*?) /wp-content/plugins/webo-site-speedup/cache/
%{HTTP_HOST}/$1/index%{ENV:WSSBR}.html%{ENV:WSSENC} [L]

```

Изменения в конфигурации nginx

Следующие правила должны быть вручную добавлены в конфигурационный файл nginx для того, чтобы включить gzip-сжатие, выставить заголовки клиентского кэширования и обеспечить работу экстремального режима серверного кэширования.

```

gzip on;
gzip_comp_level 7;
gzip_types text/plain text/xml application/xhtml+xml image/x-icon
text/css text/javascript application/javascript application/x-
javascript text/x-js text/ecmascript application/ecmascript text/
vbscript text/fluffscript image/svg+xml application/x-font-ttf
application/x-font font/opentype font/otf font/ttf application/x-
font-truetype application/x-font-opentype;

if ($http_accept_encoding ~* deflate) {
set $wssenc .df;
}
if ($http_accept_encoding ~* gzip) {
set $wssenc .gz;
}
if ($http_user_agent ~ "MSIE 6") {
set $wssbr .ie6;
}
if ($http_user_agent ~ "MSIE 7") {
set $wssbr .ie7;
}
if ($http_user_agent ~ "MSIE 8") {
set $wssbr .ie8;
}
if ($http_user_agent ~
"Android|BlackBerry|HTC|iPhone|iPod|LG|MOT|Mobile|NetFront|Nokia|Opera

```

```

Mini|Palm|PPC|SAMSUNG|Smartphone|SonyEricsson|Symbian|UP.Browser|webOS")
{
set $wssbr .ma;
}

if ($request_method != POST) {
set $cache_file_name /path-to-document-root/wp-content/plugins/
webo-site-speedup/cache/$http_host$request_uri/
index$wssbr.html$wssenc;
}
if ($http_cookie ~* "wordpress|wp-postpass_") {
set $cache_file_name '';
}
if (-f $cache_file_name) {
rewrite ^ /wp-content/plugins/webo-site-speedup/cache/
$http_host$request_uri/index$wssbr.html$wssenc break;
}

location ~* ^/wp-content/plugins/webo-site-speedup/cache/.*\.gz
{
gzip_static on;
rewrite ^(.*)\.gz$ $1;
}

location ~* ^/wp-content/plugins/webo-site-speedup/cache/.*\.df
{
gzip_static off;
add_header Content-Encoding deflate;
}

rewrite ^(.*)\.wo[0-9]+\.(css|php)$ $1.$2;
rewrite ^(.*)\.wo[0-9]+\.(js|php)$ $1.$2;
rewrite ^(.*)\.wo[0-9]+\.(jpe?g|png)$ $1.$2;

location ~* \.(jpg|jpeg|gif|png|css|js)$ {
root /path/to/document/root;
expires max;
add_header Last-Modified: $date_gmt;
}

```

4.2. Алгоритм оптимизации

Разбор содержимого

Первоначальный разбор HTML-документа

- Анализируется секция Head:
 - тэги `<script>`.
 - тэги `<style>` и тэги `<link>`.
- Выбирается содержимое JS-сценариев.
- Для содержимого head-секции и (всего набора скриптов) выполняем 3 функции (для каждой группы преобразований) — для страницы, CSS и JavaScript.

Разбор скриптов и стилей и запись их в массив

- Если у элемента есть `src` и `href`, значит, он содержит вызов внешнего файла.
- Внутренний код сразу записывается в содержимое элемента.
- Полное содержимое файлов по умолчанию выбирается, только если отключена настройка "Не проверять время изменения файлов".
- Вычисляем имя файла. Если такой файл уже существует, то больше ничего не делаем, а просто удаляем текущие стили и скрипты из head-секции и вносим туда вызовы для двух новых файлов.
 - Если подключена "ненавязчивая" загрузка JavaScript, то располагаем вызовы JavaScript-файлов перед `</body>`.
 - Если объединяем внешние скрипты и внутренний код, то располагаем финальный JavaScript-файл перед `</head>`.
 - Если внутренний код и внешние скрипты не включаем, то JavaScript-файл будет расположен сразу после CSS-файла.
 - Располагаем CSS-файл сразу за `<head>`, чтобы он загружался максимально быстро.
- Если у нас нет еще файлов в кэше и у нас включена настройка "Не проверять время изменения файлов", то содержимое файлов выбирается (чтобы создать файлы в кэше).

Оптимизация содержимого

Для JavaScript

- Все объединенное содержимое прогоняется через JSMIn / Packer или YUI Compressor.
- Если сжатие или кэширование осуществляется не через .htaccess, то формируем .php файл со всеми необходимыми gzip или кэширующими

заголовками.

Для CSS

- CSS-файлы объединяются с учетом всех конструкций @import (рекурсивно).
- К ним применяется CSS Tidy (если используются CSS Sprites или data:URI).
- Иначе просто накладываются простейшие регулярные выражения для уменьшения размера CSS-содержимого.
- Если сжатие или кэширование осуществляется не через .htaccess, то формируем .php файл со всеми необходимыми gzip или кэширующими заголовками.

Для CSS Sprites

- CSS Tidy анализирует изначальный объединенный массив CSS-файл и формирует хэш CSS-правил.
- CSS-правила анализируются на предмет наличия в них фоновых свойств (background, background-position, background-image, background-repeat).
- Дополнительно CSS-правила анализируются на предмет свойств, определяющих размер элемента (width, height, padding). Также пытаемся проанализировать CSS-селектор на наличие псевдо-вариантов (например, :hover, :link и т.д. - все по спецификации CSS3) и выяснить наследуемые свойства.
- Если у нас есть CSS-изображение с множественным background-position, то оно исключается (по-видимому, относится к уже готовому CSS Sprite).
- Если у нас есть background-repeat: repeat, то также исключаем.
- Если у нас есть background-repeat: repeat-x, и не задана высота (или задана в относительных единицах), то помечаем это изображение как repeat-xl (для включения одного такого изображения в самый низ к группе repeat-x).
- Если у нас есть background-repeat: repeat-x, и задана высота, то помечаем изображение как repeat-x.
- Если у нас есть background-repeat: repeat-y, и не задана ширина (или задана в относительных единицах), то помечаем это изображение как repeat-yl (для включения одного такого изображения в самый низ к группе repeat-y).
- Если у нас есть background-repeat: repeat-y, и задана ширина, то помечаем изображение как repeat-y.
- Если у нас есть background-position: bottom и background-repeat: no-repeat, то помечаем изображение как no-repeatb.
- Если у нас есть background-position: right и background-repeat: no-repeat, то помечаем изображение как no-repeatr.
- Если у нас есть background-repeat: no-repeat, и не задана ширина или высота элемента (или задана в относительных единицах), и background-position не right или bottom, то помечаем это изображение как no-repeati (для объединения как иконок - лесенкой).
- Если у нас есть background-repeat: no-repeat, и задана высота и ширина

элемента, и background-position не right или bottom, то помечаем изображение как no-repeat.

- Если не можем определить ни одного из указанных случаев (например, background-position : none), то исключаем изображение.
- Выясняем геометрические размеры файла изображения. Запоминаем исходное положение (background-position), увеличиваем размеры элемента на величину отступов (padding). Размеры элемента (из CSS-правил) запоминаем для вычисления исходного сдвига (всего 3 пары чисел: позиция фона, размеры изображения и возможный запас по размеру элемента).
- Объединяем все изображения согласно их типам. Для repeat-x и repeat-y ищем небольшие картинки no-repeat и располагаем их в начале файлов. В конце файлов располагаем 1 изображение repeat-xl или repeat-yl. No-repeatb объединяются "приклеенными" к низу, а no-repeatr - к правому краю итогового изображения. Остаются только группы изображений no-repeat и no-repeati.
- Для изображений no-repeat вычисляем итоговое расположение (используя двумерную матрицу, простой алгоритм свободного места).
- После создания этого спрайта вставляем в него изображения no-repeati (лесенкой, учитывая свободное место).
- Вставляем в это изображение уже подготовленные файлы no-repeatb (расширяя левый нижний угол) и no-repeatr (расширяя правый верхний угол).
- CSS-правила обновляются с учетом новых изображений и новых позиций. Несколько CSS-селекторов для одного изображения объединяются в одно правило (для оптимизации data:URI преобразования).
- Итоговые имена файлов формируются при помощи md5-хэша на основе всех CSS-правил, участвующих в образовании CSS Sprites. Перед созданием изображений проверяется их существование.
- Вызываем оптимизацию с smush.it для уменьшения размера итоговых изображений CSS Sprites.
- Если у нас не получилось создать изображение CSS Sprites (ошибка GDlib?), тогда возвращаем исходные CSS-правила для селекторов.
- Применяем data:URI (через хэш CSS-правил).

4.3. Обзор CSS Sprites

CSS Sprites являются мощным инструментом для объединения различных фоновых изображений без каких-либо ограничений при использовании в дизайне или оформлении сайта. При этом они позволяют значительно уменьшить число загружаемых объектов.

Что такое CSS Sprite?

CSS Sprite — это, фактически, одно-единственное изображение, в котором находится много различных картинок. В этом она очень похожа на карту изображений (*image map*). Мы можем при помощи CSS-свойства `background-position` вырезать из этого изображения только необходимую нам часть и показать ее в требуемом месте на странице. Также важно знать конечные размеры блока с фоновой картинкой, чтобы быть уверенным, что остальные картинки из этого CSS Sprite не будут показаны.

Как это использовать?

Необходимо собрать вместе все исходные картинки в одном файле и рассчитать их конечные позиции, чтобы включить их в CSS-правила. В случае нескольких картинок это не является сложной задачей. Однако если у вас десятки и сотни различных картинок, то это может быть весьма проблематичным и затратным по времени — корректно собрать их в одном или двух CSS Sprites.

Что здесь можно сделать?

WEBO Site SpeedUp автоматически анализирует все CSS-правила, находит возможные варианты использования CSS Sprites и применяет их. Существует ряд ограничений на корректное использование этой техники в автоматическом режиме, поэтому стоит их учитывать при создании CSS-кода для ваших сайтов. Это поможет WEBO Site SpeedUp лучше объединить изображения, избежать проблем с отображением CSS Sprites и увеличить общую производительность сайта.

Несколько правил для написания CSS-кода для улучшения создания CSS Sprites.

- Используйте только абсолютные значения для `background-position`. Использование относительных единиц (например, `em`, `%` или даже `center`, `bottom`, `right`) приведет к тому, что WEBO Site SpeedUp будет пытаться вычислить размеры блока с данной картинкой, и не всегда это можно сделать наверняка. Если нет возможности задать абсолютные значения, то постарайтесь задать абсолютные размеры для таких блоков.
- Используйте только абсолютные значения для свойств `width`, `height` и `padding`. Если WEBO Site SpeedUp находит абсолютные значения для

позиции фона, но относительные значения для размеров блока, то приложение не может корректно вычислить безопасную позицию для фоновой картинки. Поэтому оно будет объединено с другими с использованием ненужных пустых полей, либо вообще не будет использовано.

- Не используйте сложные CSS-селекторы для фоновых изображений. WEBO Site SpeedUp не анализирует наверняка все дерево CSS-правил, он может применить только несколько наиболее очевидных действий, чтобы установить требуемые свойства (`background-image`, `background-position`, `background-repeat`, `width`, `height`, `padding`). С точки зрения WEBO Site SpeedUp лучше обнаружить все эти правила в одном CSS-селекторе (никаких наследственных правил). Также это (малое количество фактических CSS-правил) является более оптимальным с точки зрения отрисовки страниц в браузерах.
- Старайтесь исключить неочевидные каскадные взаимоотношения между CSS-правилами для свойств фона. Например, вместо использования `li.item`, `.active`, `.passive` постарайтесь выставить `.menu li`, `.menu li.active`, `.menu li.passive`. Последний каскад правил будет корректно распознан WEBO Site SpeedUp.
- Исключите неиспользуемые или редко используемые CSS-правила на те страницы, где они применяются. WEBO Site SpeedUp старается объединить все обнаруженные изображения. Сказать наверняка, какие именно из этих картинок используются на данной странице, невозможно. Поэтому удаление неиспользуемых CSS-селекторов позволит уменьшить общий размер CSS Sprites (а также ускорит отображение страницы в браузерах).
- Не используйте различные значения свойства `background-position` для одного и того же изображения, если это не CSS Sprite. WEBO Site SpeedUp автоматически исключает изображения со множественными позициями фона (так как, по всей видимости, они должны быть спрайтами), поэтому такие картинки не объединяются. Для позиционирования элемента и его фона в качестве возможной альтернативы можно также использовать поля (`margin`).

Все эти правила помогут WEBO Site SpeedUp лучше проанализировать структуру вашего CSS-код и создать CSS Sprites более качественно.

Дополнительные замечания

Вы также можете использовать магическое слово 'nosprites' в ваших классах, идентификаторах или атрибутах (в коде CSS или в коде HTML относительно искомым изображениям), чтобы исключить те или иные изображения из логики создания CSS Sprites. Это будет работать как для CSS, так и для HTML Sprites. Например, если у вас есть код

```
.link .nosprites{background:url(image.png) }
```

тогда изображение `image.png` будет автоматически исключено из обработки

CSS Sprites. Также если у вас есть в коде HTML

```

```

то `image.jpg` также будет исключена из создания HTML Sprites.



4.4. API для расширений

API для расширений предназначено помочь разработчикам внедрить некоторые автоматические действия по оптимизации системы управления сайтом (как на клиентском, так и на серверном уровне) в процесс установки WEBO Site SpeedUp.

Основные возможности

Сейчас доступны следующие вызовы:

- onInstall - действия, которые будут применены при установке WEBO Site SpeedUp.
- onUninstall - действия, который будут применены при удалении WEBO Site SpeedUp.
- onBeforeOptionization - действия, которые будут применены перед проведением клиентской оптимизации.
- onAfterOptimization - действия, которые будут применены после проведения клиентской оптимизации (могут кэшироваться).
- onCache - действия, которые будут применены каждый раз перед выдачей закэшированного содержимого страниц.

Общий шаблон

Ниже приведен шаблон для файла расширения WEBO Site SpeedUp:

```
<?php
if (!class_exists('web_optimizer_plugin_joomla15')) {
/* Начинаем объявление класса */
class web_optimizer_plugin_your_cms_here {

/* Конструктор, фактически не используется */
function web_optimizer_plugin_your_cms_here() {

}

/* Установщик */
function onInstall ($root) {
/* Здесь можно использовать $root для операций с локальными
файлами сайта */
}

/* Мастер по удалению */
function onUninstall ($root) {
/* Здесь можно использовать $root для операций с локальными
файлами сайта */
}
```

```

/* пред-оптимизационные действия */
function onBeforeOptimization ($content) {
/* В $content находится текущее содержимое веб-страницы */
return $content;
}

/* пост-оптимизационные действия */
function onAfterOptimization ($content) {
/* В $content находится оптимизированное содержимое веб-страницы */
return $content;
}

/* действия при выдаче HTML-кэша */
function onCache ($content) {
/* В $content находится закешированное содержимое веб-страницы */
return $content;
}

}
/* Заканчиваем объявление класса */
}

?>

```

Более подробная информация

Имя файла, содержащего расширение, должно совпадать с именем класса, объявленного внутри него. Например, если расширение объявляет класс 'plugin_class', файл расширения должен называться 'plugin_class.php'.

Если вы не хотите использовать все доступные вызовы, можно унаследовать класс от стандартной реализации расширений - 'webo_plugin' и переопределить соответствующие методы.

[Рабочий пример для Joomla! 1.5.xx](#)

Использование

Что нужно сделать для внедрения вашего расширения:

- Разместить файл расширения (PHP-файл) в директории plugins установки WEBO Site SpeedUp.
- Добавить имя файла (без .php) вашего расширения в config.webo.php (настройка ['plugins'], самая последняя).
- Запустить установку.

Стоит отметить, что WEBO Site SpeedUp пытается автоматически определить расширения, основываясь на данных о текущей CMS. Например, для Drupal вы можете расположить файлы drupal61.php или drupal56.php в директории

plugins, и WEBO Site SpeedUp автоматически подключит их в процессе установки и добавит в config.webo.php.

4.5. Системные файлы

Ниже приведен список внутренних библиотек и файлов WEBO Site SpeedUp, которые могут располагаться вне исходной директории установки ядра.

Внутренние библиотеки

Эти файлы обеспечивают от или иной функционал, и они могут работать отдельно от файлов ядра (и поэтому при установке обычно копируются в директории кэша). Сама установка WEBO Site SpeedUp может находиться вне корневой директории сайта, поэтому вызов этих библиотек может быть не возможен.

`wo.static.php`

Первоначально располагается в папке `libs/php/`. При установке копируется в директорию CSS-кэша.

Библиотека используется для gzip-сжатия и(ли) клиентского кэширования заданных типов файлов (если в текущем серверном окружении отсутствуют необходимые модули Apache). Это хорошо известная библиотека 'статического проксирования'.

`0.gif`

Первоначально располагается в папке `libs/php/`. При установке копируется в директорию CSS-кэша.

Используется для замены исходных изображений на прозрачные при использовании HTML-спрайтов.

`yass.loader.js`

Первоначально располагается в папке `libs/js/`. При установке копируется в директорию JavaScript-кэша.

Используется для ненавязчивой загрузки всех частей JavaScript-кода в документе (содержит кросс-браузерную реализацию события `onDOMready`).

`wo.cookie.php`

Первоначально располагается в папке `libs/js/`. При установке копируется в директорию JavaScript-кэша.

Используется для проверки gzip-сжатия в тех случаях, когда не передается корректный заголовок `Accept-Encoding` (PHP-файл отправляет сжатый JavaScript-код, который выставляе тсоответствующие cookie).

`web.optimizer.stamp.png`

Первоначально располагается в папке `images/`. При установке копируется в

директорию CSS-кэша.

Используется в качестве водяного знака WEBO Site SpeedUp для некоммерческой редакции (и может быть использован в любой другой редакции WEBO Site SpeedUp).

Файлы достижений

Эти файлы необходимы для отображения текущих достижений в ускорении сайта.

webo-site-speedup.back.jpg

Первоначально располагается в папке `libs/css/`. При первом обращении к странице Достижений копируется в директорию CSS-кэша.

Используется для отображения общего (оранжевого) фона на странице достижения ускорения сайта.

webo-site-speedup.rocket.png

Первоначально располагается в папке `libs/css/`. При первом обращении к странице Достижений копируется в директорию CSS-кэша.

Содержит все части графики достижений для отображения финальной схемы ракеты.

webo-site-speedup88.png

При каждом заходе на страницу Достижений сверяется с последней версией на сервере WEBO Software.

Содержит маленькое (88x88) изображение достижений при ускорении сайта.

webo-site-speedup125.png

При каждом заходе на страницу Достижений сверяется с последней версией на сервере WEBO Software.

Содержит небольшое (125x125) изображение достижений при ускорении сайта.

webo-site-speedup161.png

При каждом заходе на страницу Достижений сверяется с последней версией на сервере WEBO Software.

Содержит среднее (161x161) изображение достижений при ускорении сайта.

webo-site-speedup250.png

При каждом заходе на страницу Достижений сверяется с последней версией на сервере WEBO Software.

Содержит большое (250x250) изображение достижений при ускорении сайта.

webo-site-speedup.php

При каждом заходе на страницу Достижений пересоздается для самого последнего состояния приложения.

Содержит весь HTML-код страницы достижений при ускорении сайта.

webo-site-speedup.css

При каждом заходе на страницу Достижений пересоздается для самого последнего состояния приложения.

Содержит всю CSS-часть (mhtml) страницы достижений при ускорении сайта.

Временные файлы

Эти файлы обычно не присутствуют в папках кэша, но в некоторых случаях (во время выполнения административной логики WEBO Site SpeedUp) могут там появиться.

progress.html

Может быть обнаружен внутри папки JavaScript-кэша.

Содержит текущую стадию оптимизации или номер файла при загрузке обновлений.

htaccess.test

Может быть обнаружен внутри папки JavaScript-кэша.

Содержит (пустой) текст из файла `libs/php/css.sprites.php` (при загрузке через curl текст будет пустым). Если в этом файле находится какой-либо текст (ошибка Apache), значит, для сайта можно использовать `.htaccess`. Обычно такая проверка на возможность использования `.htaccess` вызывает запись об ошибке в лог-файлах. Не стоит беспокоиться по этому поводу.

module.test

Может быть обнаружен внутри папки JavaScript-кэша.

Содержит текст файлов `libs/js/wo.cookie.php`, `libs/js/yass.loader.php` или `libs/js/yass.loadbar.js` (зависит от того, какой модуль проверяется). Данная процедура используется для проверки модулей Apache (если вызов `apache_get_modules` ничего не возвращает).

4.6. Алгоритм настройки

Автоматическая настройка — Проверка окружения

1. **Отключение WEBO Site SpeedUp и всех его настроек** (для корректной проверки всех комбинаций настроек без опасности нарушить работоспособность сайта).
 - Включаем «Добавлять <!--wss--> в body оптимизированных страниц».
2. **Клиентское кэширование** (выставление заголовков Cache-Control и Expires в далекое будущее для предотвращения повторного запроса статических ресурсов с сайта).
 - Включаем «Кэшировать файлы CSS на клиенте».
 - Включаем «Кэшировать файлы JavaScript на клиенте».
 - Включаем «Кэшировать изображения на клиенте».
 - Включаем «Кэшировать файлы шрифтов на клиенте».
 - Включаем «Кэшировать видео-файлы на клиенте».
 - Включаем «Кэшировать остальные файлы на клиенте».
 - **Серверное кэширование** (снижение серверных издержек на создание HTML-страниц).
 - Проверяем текущую задержку при создании главной страницы (если есть библиотека curl).
 - Включаем серверное кэширование (если WEBO Site SpeedUp установлен как системное расширение и задержка больше 500 мс).
 - **Проверка .htaccess** (использование модулей Apache для технологий клиентской и серверной оптимизации).
 - Проверяем, есть ли возможность использовать SymLinks.
 - Загружаем список доступных модулей Apache.
 - mod_gzip
 - mod_deflate (+ mod_filter)
 - mod_expires
 - mod_headers
 - mod_rewrite
 - mod_mime
 - Включаем доступные модули через .htaccess.
 - Проверяем, доступен ли сайт (если есть библиотека curl).
 - Если сайт не доступен, то отключаем использование всех модулей Apache.

Автоматическая настройка — Оптимизация CSS

1. Объединение файлов (уменьшение числа запросов к CSS-файлам).

- Вычисляем количество тегов (стилей) `<link>` и `<style>` внутри `<body>`.
- Если число стилей больше нуля, то включаем «Объединять CSS, подключенный в тэге `<head>` и `<body>`»
- Иначе включаем «Объединять только CSS, подключенный в тэге `<head>`»
- Включаем «Объединять встроенный код CSS»
- Включаем «Объединять внешние файлы CSS» (если есть библиотека curl).
- Сравниваем стили для основных элементов оптимизированной страницы и страницы без ускорения.
- Если стили различаются, то отключаем «Объединять встроенный код CSS»
- Сравниваем стили снова, если стили различаются, то отключаем «Объединять внешние файлы CSS».
- Сравниваем стили снова, если стили различаются, то отключаем «Объединять только CSS, подключенный в тэге `<head>`».
- **Минимизация файлов** (уменьшение размера CSS-кода).
- Включаем «Минимизировать CSS».
- Сравниваем стили, если стили различаются, то отключаем «Минимизировать CSS».
- **gzip-сжатие файлов** (уменьшение размера CSS-кода).
- Включаем «Применять gzip для CSS».
- Сравниваем стили, если стили различаются, то «Применять gzip для CSS».

Автоматическая настройка — Оптимизация JavaScript

1. Объединение файлов (уменьшение числа запросов к JavaScript-файлам).

2. Включаем «Объединять только JavaScript, подключенный в тэге `<head>`».

- Загружаем страницу без ускорения еще раз и сравниваем встроенный код JavaScript с первоначальным.
- Если встроенный код не отличается, то включаем «Объединять встроенный код JavaScript».
- Включаем «Объединять внешние файлы JavaScript» (если есть библиотека curl).
- Включаем «Переносить вызов объединенного кода JavaScript перед `</head>`».
- Получаем число JavaScript-ошибок на оптимизированной странице.
- Если ошибки есть, то отключаем «Объединять встроенный код JavaScript».
- Проверяем ошибки снова, если они есть, то отключаем «Переносить вызов объединенного кода JavaScript перед `</head>`».

- Проверяем ошибки снова, если они есть, то отключаем «Объединять внешние файлы JavaScript» и включаем «Переносить вызов объединенного кода JavaScript перед </head>».
- Проверяем ошибки снова, если они есть, то последовательно добавляем в настройку «Исключать из объединения JavaScript файлы» файлы с исходной страницы.
- Проверяем ошибки снова, если они есть, то отключаем «Объединять только JavaScript, подключенный в тэге <head>».
- **Минимизация файлов** (уменьшение размера JavaScript-кода).
- Включаем «Минимизировать с помощью JSMIn».
- Получаем число JavaScript-ошибок на оптимизированной странице.
- Если ошибки есть, то отключаем «Минимизировать с помощью JSMIn» и включаем «Минимизировать с помощью YUI Compressor».
- Проверяем ошибки снова, если они есть, то отключаем «Минимизировать с помощью YUI Compressor» и включаем «Минимизировать с помощью Packer».
- Проверяем ошибки снова, если они есть, то отключаем «Минимизировать с помощью Packer».
- **Удаление дубликатов** (уменьшение размера JavaScript-кода).
- Проверяем наличие дублирующихся скриптов на исходной странице.
- Если есть дубликаты, то включаем «Удалять дубликаты».
- Получаем число JavaScript-ошибок на оптимизированной странице.
- Если ошибки есть, то отключаем «Удалять дубликаты».
- **gzip-сжатие файлов** (уменьшение размера JavaScript-кода).
- Включаем «Применять gzip для JavaScript».
- Получаем число JavaScript-ошибок на оптимизированной странице.
- Если ошибки есть, то отключаем «Применять gzip для JavaScript».

Автоматическая настройка — Оптимизация HTML

1. **Минимизация файлов** (уменьшение размера HTML-кода).
 - Включаем «Минимизировать HTML».
 - Сравниваем изменение HTML (размер DOM-дерева для тега head и размер встроенного кода JavaScript в тэге body) для оптимизированной и страницы без ускорения.
 - Если HTML изменился, то отключаем «Минимизировать HTML».
 - **Удаление комментариев** (уменьшение размера HTML-кода).
 - Включаем «Удалять комментарии из HTML».
 - Сравниваем изменение HTML для ускоренной и исходной страницы.
 - Если HTML изменился, то отключаем «Удалять комментарии из HTML».
 - **gzip-сжатие** (уменьшение размера HTML-кода).

- Включаем «Применять gzip для HTML».
- Включаем «Применять gzip для шрифтов».
- Включаем «Проверять поддержку gzip при помощи cookie».
- Включаем «Применять deflate вместо gzip для IE6, IE7».
- Загружаем оптимизированную страницу и пытаемся прочитать ее содержимое.
- Если содержимое не читается, то отключаем «Применять gzip для HTML», «Применять gzip для шрифтов», «Проверять поддержку gzip при помощи cookie» и «Применять deflate вместо gzip для IE6, IE7».

Автоматическая настройка — Производительность

1. Проверка изменения файлов (уменьшение издержек при оптимизации).
2. Включаем «Не проверять время изменения файлов».
 - Включаем «Не применять регулярные выражения».
 - Сравниваем изменение HTML для ускоренной и исходной страницы.
 - Если HTML изменился, то отключаем «Не применять регулярные выражения».

Автоматическая настройка — data:URI

1. Фоновые изображения (уменьшение числа HTTP-запросов).
 - Включаем «Использовать data:URI».
 - Включаем «Использовать mhtml».
 - Включаем «Отделять изображения от кода CSS».
 - Включаем «Загружать изображения по событию DOMReady».
 - Сравниваем стили для основных элементов оптимизированной страницы и страницы без ускорения.
 - Если стили различаются, то отключаем «Использовать data:URI», «Использовать mhtml», «Отделять изображения от кода CSS» и «Загружать изображения по событию DOMReady».
 - Загружаем второй CSS-файл (содержащий фоновые изображения) с оптимизированной страницы и получаем его размер.
 - Если размер файла меньше 102400 байт, то отключаем «Отделять изображения от кода CSS» и «Загружать изображения по событию DOMReady».

Автоматическая настройка — CDN

1. Распределение файлов JavaScript (увеличение параллельности запросов).
 - Вычисляем хост файлов JavaScript на странице без ускорения, который включает основной домен сайта.



- Если хост не равен основному домену (является поддоменом), то записываем его в настройку «Хост для JavaScript-файла(-ов)» и включаем «Распределять JavaScript-файлы»
- **Распределение файлов CSS** (увеличение параллельности запросов).
- Вычисляем хост файлов CSS на странице без ускорения, который включает основной домен сайта.
- Если хост не равен основному домену (является поддоменом), то записываем его в настройку «Хост для CSS-файла(-ов)» и включаем «Распределять CSS-файлы»
- **Распределение изображений** (увеличение параллельности запросов).
- Вычисляем хост(ы) изображений на странице без ускорения, который включает основной домен сайта.
- Если хост(ы) не равен основному домену (является поддоменом), то записываем его в настройку «Использовать хосты (через пробел)» и включаем «Распределять изображения».

Автоматическая настройка — HTML Sprites

1. **HTML Sprites** (уменьшение числа HTTP-запросов).
2. Вычисляем число изображений на странице без ускорения, размер которых меньше 16x16
 - Если число изображений более 2, то включаем «Объединять HTML-изображения», «Объединять изображения только для текущей страницы», а «Максимальная ширина и высота HTML-изображений» выставляем в 16.

Автоматическая настройка — Ненавязчивый JavaScript

1. **Неблокирующий JavaScript** (ускорение отображение страницы за счет перемещения вызовов JavaScript).
2. Включаем «Переносить вызов объединенного JavaScript-файла перед `</body>`».
- Получаем число JavaScript-ошибок на оптимизированной странице.
- Если ошибки есть, то отключаем «Переносить вызов объединенного JavaScript-файла перед `</body>`».
- **Отложенная загрузка** (ускорение отображения страницы).
- Вычисляем количество известных виджетов на странице.
- Если виджеты есть, то включаем «Перенести вызовы JavaScript-информеров перед `</body>`».
- Вычисляем количество известных рекламных блоков на странице.

- Если рекламные блоки есть, то включаем «Перенести рекламные вызовы JavaScript перед `</body>`».
- Вычисляем количество фреймов на странице, которые не относятся к рекламным блокам и виджетам.
- Если фреймы есть, то включаем «Откладывать загрузку `iframe`».

4.7. Шаблоны ненавязчивого JavaScript

Все шаблоны для отложенной загрузки JavaScript-кода (с целью предотвращения блокирования отрисовки страницы) расположены в конфигурационном файле `libs/php/config.unobtrusive.php` внутри ядра WEBO Site SpeedUp. Таким образом, вы можете легко добавлять новые шаблоны (или даже удалять старые) без необходимости модификации сложной логики приложения.

Описание шаблона

У каждого шаблона есть 7 параметров:

- Группа (используется для соответствия текущей настройке в конфигурации).
- Идентификатор (должен быть уникальным, используется для формирования ID для блоков источника/назначения в макете страницы).
- Маркер, `marker` (короткая строка в HTML-коде, позволяющая определить, нужно ли применять регулярное выражение для блока или нет).
- Регулярное выражение, `regexr` (основной шаблон для перемещения JavaScript-блока из его первоначального положения в самый конец документа).
- `Onload_before` (регулярное выражения, шаблон совпадения для поиска JavaScript-кода и применения к нему техники продвинутой отложенной загрузки, на событие `window.onload`).
- `Onload_after` (регулярное выражения, шаблон подстановки для включения найденного JavaScript-кода в документ после того, как отработал шаблон `onload_before`).
- Высота, `height` (используется для предотвращения перерисовки страницы для динамическом изменении содержания целевого контейнера, фактически, динамического появления отложенного JavaScript-блока).

Примеры

Отложенная загрузка JavaScript-кода

Давайте возьмем для примера шаблон Amazon ads. Он представлен следующим шаблоном отложенной загрузки:

```
'aa' => array(
  'marker' => 'amazon_ad',
  'regexr' => "<script[^>]+><!--[^\da-zA-Z]*amazon_ad_tag.*?ads.js\"></script>"
)
```

Как это работает? Если HTML-документ соответствует заданной строке, маркеру - `amazon_ad` - то к документу применяется поиск по регулярному выражению

(**regexp**), и выбираются все блоки JavaScript-кода, ему соответствующие. Затем все эти блоки кода последовательно заменяются на

(контейнеры назначения, **aa** является уникальным идентификатором блока среди всех шаблонов отложенной загрузки). А в конец документа добавляется эти самые блоки кода, заключенные в (контейнеры-источники). После того как весь JavaScript-код (для соответствующего блока) загружается в контейнер-источник, все содержимое источника (в котором может быть очень много различного кода) перемещается в контейнер назначения. В макете документа не будет происходить никаких изменений, если задан параметр **height** (высота блока). В этом случае только динамическое содержание (обычно рекламное) появляется на странице внутри заданного контейнера. В результате у нас нет блокирующего JavaScript, но весь сторонний контент (который поддерживается шаблонами в конфигурации) загружается правильно.

В большинстве случаев этого шаблона достаточно для существенного ускорения загрузки сайта.

Улучшенный режим перемещения JavaScript

Если вы хотите отложить загрузку, например, виджета Google Search к событию **window.onload**, то просто добавить его загрузку в обработчик указанного события не получится. Будет необходимо эмулировать поведение **document.write** для создания новых узлов DOM-структуры с заданным исходным кодом виджета. Для этой цели используются два расширенных параметра шаблона отложенной загрузки - **onload_before** и **onload_after**.

Давайте рассмотрим следующий пример:

```
'gs' => array(
'marker' => 'setOnLoadCallback',
'regexp' => "<script src=\"https://www.google.com/jsapi\"
type=\"text/javascript\">" .
"[\r\n\s\t]*</script>[\r\n\s\t]*<script type=\"text/
javascript\">(//\s*<![CDATA\[]?)" .
"[\r\n\s\t]*google\.load\(['\"]search.*?</script>",
'onload_before' =>
'.*?google.load\(\s*['\"]search['\"](.*)\);(.*)google.setOnLoadCallback['
.
'\r\n\s\t]*\((function\(\)\{(.*)\},\strue\);(.*)</script>',
'onload_after' => 'document.write(\'\x3cscript
src="//www.google.com/jsapi" type="text/javascript">'
'\x3c/script>\');setTimeout(function(){if(typeof
google!=="undefined"&&typeof google.load!=="'
'"undefined"){google.load("search"$1);setTimeout(function(){if(typeof
google.search!=="undefined"'
'&&typeof
google.search.CustomSearchControl!=="undefined"){${2}${3}${4};setTimeout(function()
```

```
'var a=document.forms,b=0,c;while(c=a[b++]){if(c.className=="gsc-  
search-box"){wss_onload_ready=1}}' .  
'if(!wss_onload_ready){setTimeout(arguments.callee,20)}},20)}else{setTimeout(  
.  
,10)}else{setTimeout(arguments.callee,10)}},10);'  
)
```

Оба регулярных выражения используются следующим образом:

```
$document = preg_replace("@ " . $onload_before . "@is",  
$onload_after, $document);
```

Вы можете задать внутри первого выражения (**onload_before**) любые группы для дальнейшего использования во втором выражении. Будьте также аккуратны с использованием динамической логики и применяйте **setTimeout** для определения того момента, загрузилась ли требуемая библиотека для применения обработчика.

Переменная JavaScript **wss_onload_ready** должна быть выставлена в 1 после того, как закончилась загрузка (закончился последний обработчик загрузки) данного блока. Это необходимо для начала загрузки следующего блока, объявленного таким же образом, и корректной работы эмуляции **document.write**.

5. Устранение неполадок и поддержка

Используйте дополнение Firebug для Firefox (или аналогичные средства), чтобы просматривать код веб-страниц, диаграммы загрузки объектов веб-страниц, заголовки запросов серверу и ответов от него. Эта сведения помогут вам получить максимум информации о работе WEBO Site SpeedUp на вашем сайте.

Если вы столкнулись с проблемой во время установки или работы с WEBO Site SpeedUp, обратитесь к следующим разделам:

- Проблемы при установке
- Проблемы на стороне клиента
- Проблемы на стороне сервера

Если ни один из этих разделов не помог решить вашу проблему пожалуйста обратитесь к разделу технической поддержки.

5.1. Проблемы при установке

Отсутствуют права на запись

Для корректной работы WEBO Site SpeedUp следующие файлы и директории должны иметь права на запись.

- Файл `index.php` в корневой директории сайта. В этот файл чаще всего дописываются вызовы, интегрирующие WEBO Site SpeedUp с текущей системой.
- Файл `.htaccess` в корневой директории сайта. Если же этот файл отсутствует, необходимы права на запись в саму корневую директорию, чтобы файл мог быть создан. В файле `.htaccess` содержатся инструкции, необходимые для работы многих методов ускорения сайта.
- Директория, в которую установлен WEBO Site SpeedUp и все вложенные файлы. В этой директории сохраняются файлы обновлений WEBO Site SpeedUp, настройки пользователей.
- Директории кэша HTML, CSS и JavaScript и все вложенные файлы. В этих директориях содержатся оптимизированные файлы — результат работы WEBO Site SpeedUp. По умолчанию эти директории расположены внутри директории WEBO Site SpeedUp, но их расположение может быть изменено пользователем.

Сайт отображается неправильно, но WEBO Site SpeedUp отключен или даже удален

Неверные директивы в `.htaccess`

В корневой директории сайта находится файл `.htaccess`, который во многих случаях изменяется после установки WEBO Site SpeedUp. Там же находится копия оригинального файла с именем `.htaccess.backup`. Восстановите оригинальный файл или убедитесь, что в файле `.htaccess` не осталось директив, созданных WEBO Site SpeedUp. Такие директивы всегда окружены двумя строками:

```
[какие-либо директивы]
```

Активен клиентский или серверный кэш

Кэшированные страницы могут обращаться к несуществующим ресурсам или содержать код, заставляющий сайт отображаться неправильно. Найдите все решения, которые используются для кэширования и очистите кэш, чтобы убедиться в том, что вы видите актуальные версии страниц.

Восстановление пароля WEBO Site SpeedUp

Если WEBO Site SpeedUp установлен в качестве отдельного приложения, откройте директорию его установки, найдите файл `config.webo.php` и удалите значение параметра `password` на строке 7, чтобы строка выглядела так:

```
$compress_options['password'] = "";
```

После этого откройте WEBO Site SpeedUp в браузере и вновь введите ваши регистрационные данные включая пароль. Все конфигурации и файлы кэша при этом будут сохранены.

Если WEBO Site SpeedUp установлен в качестве плагина, пароль может быть восстановлен стандартным для данной системы способом.

Функционирование WEBO Site SpeedUp

Довольно часто приходится разобраться с тем, работает ли WEBO Site SpeedUp на сайте, или его установка каким-то образом не подключилась к обработке HTML-документа. Начиная с версии 0.9.0 это можно установить, найдя строку `<!--wss-->` в коде страницы (если метка там отсутствует, значит WEBO Site SpeedUp не обрабатывает). Поскольку эту метку можно отключить в конфигурации, будьте, пожалуйста, внимательны с этим аспектом.

Для более ранних версий это решение можно принять на основе отсутствия в коде HTML-документа отступов в начале строки, двойных переводов строк или наличия характерных заэкшированных имен файлов в head-секции (`cache/1234a6789b.css` или `cache/1234c6789d.js`, здесь `1234c6789d` — произвольная строка в шестнадцатеричной записи).

Если обнаружить следы работы WEBO Site SpeedUp не удалось, то необходимо перепроверить корректность вызовов WEBO Site SpeedUp в файлах системы управления сайтом и, возможно, провести установку приложения еще раз — таким образом WEBO Site SpeedUp сможет самостоятельно произвести все необходимые изменения.

Для того, чтобы получить информацию о необходимых изменениях для установки WEBO Site SpeedUp как отдельного приложения вы можете открыть страницу *Состояние системы* и перейти на вкладку *Установка и удаление*.

Также стоит убедиться, что сам WEBO Site SpeedUp активен в настройках конфигурации (для всех типов установки кроме плагина для Joomla!). Для этого необходимо проверить в файле `web-optimizer/config.webo.php` следующую строку

```
$compress_options['active'] = "1";
```

И выставить для нее значение в 1 (если оно равно 0). Это активирует WEBO Site SpeedUp для вашего сайта.

Если и это не помогло, то, пожалуйста, проверьте настройки остальных

плагинов и дополнений, которые могут применять gzip к содержимому (из-за чего WEBO Site SpeedUp может получать все данные в сжатом виде и не может их проанализировать). Попробуйте отключить gzip в каждом таком плагине или глобальных настройках системы, чтобы сжатие не распространялось на HTML-код.

Неверное вычисление путей

Иногда у WEBO Site SpeedUp не получается правильно вычислить все пути по умолчанию для проведения установки (это может выражаться в отсутствии стилей или скриптов на сайте, а также в фатальной PHP-ошибке после установки). Обычно это происходит на CGI-окружениях. WEBO Site SpeedUp пробует следующие подходы для вычисления путей.

- Обнаруживает корень хоста (document root). Обычно он определен в `$_SERVER`ом случае, если корень хоста обнаружить не удалось (или он определен в недоступную папку), WEBO Site SpeedUp пробует получить значения переменной окружения `SCRIPT_FILENAME` и исключает из нее значение другой переменной — `SCRIPT_NAME` (это обычно помогает исправить ситуацию).
- В том случае если сайт установлен в отдельной директории внутри хоста (например, в папке внутри document root), у WEBO Site SpeedUp есть отдельная настройка для корня сайта (не хоста). Обычно она также вычисляется автоматически.
- Если CGI-окружение настроено не совсем верно (например, не определена переменная `$_SERVER['DOCUMENT_ROOT']`, а переменные окружения `SCRIPT_NAME`, `SCRIPT_FILENAME` определены относительно внутреннего PHP-процесса), WEBO Site SpeedUp не удастся правильно вычислить корень хоста. Что выражается в неверном вычислении всех остальных директорий (в большинстве случаев они автоматически определяются после корня хоста, document root).

В последнем случае необходимо определить все пути для WEBO Site SpeedUp вручную. Это можно сделать через интерфейс WEBO Site SpeedUp: все пути обычно приведены в самом начале параметров для изменения.

5.2. Проблемы на стороне клиента

Внимание! После каждого внесения изменений вам необходимо обновить кэш, для того чтобы изменения вступили в силу.

Ошибка браузера, или некорректные символы вместо сайта, или Content Encoding Error

Обычно это связано с двойным сжатием HTML-документа. Одно из сжатий может быть наложено самим WEBO Site SpeedUp, а второе — как используемой системой управления сайтом, так и сервером.

Чтобы снять одно из накладываемых сжатий, можно отключить его либо в CMS, либо в конфигурации сервера, либо в самом WEBO Site SpeedUp:

```
Gzip-архивирование → Применить gzip для HTML → Нет
```

Белый экран вместо сайта или слишком большое время ожидания

Основная проблема возникновения белого экрана (превышение лимита памяти при создании CSS Sprites) уже устранена в версии 0.5+, поэтому если у вас более старая версия, то стоит просто ее обновить. В любом случае вы можете попробовать отключить CSS Sprites полностью:

```
CSS Sprites → Применять CSS Sprites → Нет
```

а затем последовательно увеличивать максимальную высоту и ширину изображений

```
CSS Sprites → Максимальная ширина и высота изображений (в px) →  
100 ... 900
```

или же исключить ряд файлов из объединения

```
CSS Sprites → Исключать из CSS Sprites файлы → Список файлов через  
пробел
```

Кроме этого, можно попробовать отключить объединение HTML-изображений

```
CSS Sprites → Объединять HTML-изображения → Нет
```

Также стоит заглянуть в логи ошибок вашего сервера, чтобы узнать, что привело к такому состоянию. Обычно это помогает решить проблему.

В некоторых случаях белый экран возникает из-за некорректной установки приложения либо двойного сжатия. Как решить эти проблемы, описано чуть выше.

Если все приведенные шаги не принесли результата, то можно попробовать отключить часть настроек WEBO Site SpeedUp, чтобы понять, какие вещи ваш

сервер может выполнить самостоятельно. Начать стоит с групп «Объединение CSS», «Объединение JavaScript», «Минимизация» и «Gzip-архивирование» и двигаться в сторону подключения более детальных параметров (например, в группах «Клиентское кэширование», «.htaccess», «CSS Sprites» или «Использование data:URI»).

Некорректный внешний вид

Это может быть причиной различных проблем, но для начала можно попробовать отключить CSS Sprites:

CSS Sprites → Применить CSS Sprites → Нет

затем, если это не помогло ситуации, можно отключить data:URI:

Использование data:URI → Использовать data:URI → Нет

После этих шагов все подключаемые CSS-файлы не будут обрабатываться через CSS Tidy, а будут только объединяться (и будет выполняться первичная минимизация).

Если и после этого внешний вид сайта «разъехался», то стоит отключить объединение внешних файлов

Объединение CSS → Включить объединение внешних CSS-файлов → Нет

и(ли) внутренних стилей

Объединение CSS → Объединять встроенный CSS-код → Нет

или вообще минимизацию для CSS-файлов

Объединение CSS → Объединение CSS-файлов → Не объединять CSS-файлы

Если есть желание разобраться в возникшей проблеме более детально, то можно при отключенном объединении стилей внутри head попробовать исключить один за другим при помощи следующей настройки:

Объединение CSS → Исключать из объединения CSS-файл → Список файлов через пробел

Найти тот файлы (или те файлы), которые обрабатываются некорректно, привести их к стандартному состоянию при помощи валидатора jigsaw.w3.org/css-validator/ и попробовать объединить снова.

В конце концов, есть настройка

Объединение CSS → Включать в объединенные CSS-файлы дополнительный код

с помощью которой можно определить дополнительные стили, которые будут подключаться после всего объединенного кода. Это позволяет гарантированно устранить любые ошибки объединения.

Пропавшие или неправильные фоновые изображения

В некоторых случаях проблемы после установки WEBO Site SpeedUp сводятся к тому, что некоторые фоновые изображения пропадают или «портятся». Для решения этого набор проблем необходимо установить, с какими первоначальными фоновыми изображениями возникают трудности (используя любые средства для отладки верстки) и исключить эти изображения из процесса создания CSS Sprites:

CSS Sprites → Исключить из CSS Sprites файлы → Список файлов через пробел

Если данная мера не приносит результата, то CSS Sprites можно вообще выключить:

CSS Sprites → Применить CSS Sprites → Нет

Также возможно, что стоит воспользоваться настройкой

Объединение CSS → Включать в объединенные CSS-файлы дополнительный код

чтобы определить каскад стилей более точно, так как возможно проблемы отображения при использовании настройки

Использование data:URI → Отделить картинки от CSS-кода

Недоступность стилей, скриптов и изображений при использовании клиентского кэширования и веб-сервера nginx

В WEBO Site SpeedUp реализованы два способа сброса клиентского кэша:

- С помощью сменяющихся GET-параметров, например: 3a1a23c4ea.js?wo1285789196. При этом некоторые прокси-серверы могут игнорировать GET-параметр и даже при его изменении по-прежнему отдавать старый файл.
- С помощью сменяющихся суффиксов, например: 3a1a23c4ea.wo1285789196.js. Это наиболее предпочтительный способ, но тут необходимо перенаправлять все запросы к файлам с такими суффиксами на реальные, существующие файлы. И поскольку nginx не поддерживает динамическое изменение конфигурации, WEBO Site SpeedUp не может автоматически создать необходимые правила для перенаправления файлов.

Решить эту проблему можно одним из следующих способов:

- Отключить опцию `.htaccess` → Использовать `mod_rewrite`. В этом случае для сброса клиентского кэша будет использоваться первый способ к

запросам закешированных файлов будут добавляться уникальные GET-параметры.

- Отключить опцию `.htaccess` → Использовать `mod_expires` (при включенном кэшировании для статических файлов) или опцию `.htaccess` → Использовать `mod_deflate` (при включенном сжатии статических файлов в WEBO Site SpeedUp). В этом случае расширения файлов статических ресурсов будут изменены на `.php`, а, соответственно, сжатие или заголовки клиентского кэширования будут применяться динамически при запросе каждого такого файла. В результате будут запрашиваться файлы вида: `3a1a23c4ea.wo1285789196.php`, которые будут обслуживаться напрямую сервером Apache (с работающим модулем `mod_rewrite`), минуя `nginx`.
- В конфигурацию `nginx` можно вручную прописать правила перенаправления, аналогичные тем, которые автоматически добавляются в `.htaccess` для Apache. Выглядеть они могут примерно так:

```
rewrite ^(.*)\.wo[0-9]+\.(css|js)$ $1.$2 break;
```

Желательно также максимально задействовать `nginx` для работы со статическими ресурсами и реализовать клиентское кэширование и сжатие с его помощью. Идеологически это наиболее верный способ, но он требует ручного изменения конфигурации `nginx`.

Некорректное поведение сайта

Обычно это сводится к тому, что часть клиентской логики перестает отрабатывать. Нужно хорошо понимать, что если, например, форма для ввода комментариев у вас на сайте выводится при помощи JavaScript, и после WEBO Site SpeedUp она перестала выводиться, то это проблемы не внешнего вида, а клиентской логики.

Стоит начать решение данной проблемы с отключения настройки

Объединение JavaScript → Безопасный режим объединения → Нет

В некоторых случаях режим объединения может быть настолько безопасен, что будет конфликтовать с текущим набором клиентских скриптов.

Для локализации проблемы с клиентской логикой рекомендуется попробовать отключить минимизацию JavaScript-файлов:

Минимизация → Минимизация JavaScript-файлов → Не минимизировать JavaScript

а потом (при сохранении проблемы) попробовать исключить один за другим отдельные файлы:

Объединение JavaScript → Исключать из объединения JavaScript-файлы → Список файлов через пробел

При установлении имени файла, на котором объединение отказывает, можно

просто его исключить из общего пакета либо попытаться настроить его корректное включение в первоначальную логику (обычно ошибки происходят из-за некорректного синтаксиса исходных файлов, которые не конфликтуют внутри одного окружения-файла, но начинают конфликтовать при объединении этих окружений).

Дополнительно можно попробовать исключить из объединения JavaScript-код

Объединение JavaScript → Объединять встроенный JavaScript-код → Нет

и расположить объединенный JavaScript-файл около начала страницы

Объединение JavaScript → Переносить вызов объединенного JavaScript-файла перед </head> → Нет

Недоступность файлов для пользователей

Иногда возникают проблемы с отображением и функционированием сайта у всех пользователей, хотя у владельца сайта (в его браузере) при этом все хорошо. Если диагноз проблемы звучит подобным образом, то нужно проверить, по какому адресу находятся директории кэширования, сам WEBO Site SpeedUp и включена ли защита WEBO Site SpeedUp от внешнего доступа:

Состояние системы → Параметры → Защитить установку Веб Оптимизатора с помощью htpasswd → Да

В этом случае директории кэширования нужно вынести из папки самого WEBO Site SpeedUp (например, в корневую директорию cache, доступную на запись для веб-сервера) либо отключить защиту приложения:

Состояние системы → Параметры → Защитить установку Веб Оптимизатора с помощью htpasswd → Нет

Иконка WEBO Site SpeedUp появилась на сайте

WEBO Site SpeedUp помещает свою иконку в файл favicon.ico в корневой директории сайта тогда, когда не может обнаружить там другой иконки или не находит ссылки на какие-либо другие иконки в коде страниц. Эта мера позволяет браузерам загрузить иконку и закэшировать ее на долгое время. Иначе браузеры будут получать 404 ошибку при каждом последующем запросе. Если вы хотите установить собственную иконку, просто удалите файл favicon.ico (и его сжатую версию favicon.ico.gz) из корневой директории сайта. Спустя какое-то время браузеры обновят свой кэш и вы увидите вашу новую иконку.

5.3. Проблемы на стороне сервера

Internal Server Error или белая страница после включения WEBO Site SpeedUp

WEBO Site SpeedUp не изменяет исходные коды сайта, но может улучшать конфигурацию Apache при помощи файла `.htaccess`, что иногда может привести к проблемам.

Чтобы полностью отключить WEBO Site SpeedUp и восстановить исходное состояние сайта просто отключите модуль WEBO Site SpeedUp в его панели управления (НЕ УДАЛЯЙТЕ и НЕ ОТКЛЮЧАЙТЕ модуль в админке сайта до этих пор).

Если админка сайта недоступна:

- Восстановите файл `.htaccess` в корневой директории сайта при помощи резервной копии `.htaccess.backup` созданной WEBO Site SpeedUp. Если резервной копии вы не обнаружили, просто удалите все инструкции добавленные WEBO Site SpeedUp в текущий файл `.htaccess`.
- Найдите файл `/web-optimizer/config.webo.php` и измените значение параметра `$compress_options['active']` с "1" на "0".

Если и после этого проблема сохранилась (что крайне маловероятно), проанализируйте логи ошибок сервера (вы можете запросить их у хостера) чтобы понять и исправить проблему.

Потребление серверных ресурсов WEBO Site SpeedUp

Возможны три различных ситуации, в каждой из которых нагрузка на систему отличается.

- WEBO Site SpeedUp установлен, но отключен (т.е. находится в режиме отладки). В этой ситуации абсолютно никакая дополнительная нагрузка на систему не ложится.
- WEBO Site SpeedUp включен и его кэш полностью заполнен. В этой ситуации нагрузка минимальна, практически незначительна, и не должна превышать 5-10 миллисекунд.
- WEBO Site SpeedUp включен, но его кэш не заполнен. В этой ситуации нагрузка может быть как небольшой, так и значительной. Точное количество серверных издержек зависит от серверного окружения и конфигурации WEBO Site SpeedUp. Ниже эта ситуация объяснена более подробно.

Если вы наблюдаете высокую нагрузку на процессор и память, кэш WEBO Site SpeedUp был недавно обновлен, пожалуйста, подождите некоторое время. В

большинстве случаев потребуется не больше часа для того, чтобы все необходимые файлы кэша были созданы и отдавались без дополнительной нагрузки на систему. Это происходит естественным образом при посещении вашего сайта пользователями или поисковыми роботами.

Если кэш был обновлен достаточно давно, а нагрузка на сервер по-прежнему высока, возможно, некоторые файлы кэша создаются при просмотре каждой страницы сайта. Такую ситуацию легко отследить открыв вкладку *Кэш* на станции *Состояние системы*. Если размер кэша растет после просмотра каждой страницы, то возможно, что к этому приводят некоторые опции WEBO Site SpeedUp.

Для примера, когда на каждой странице находится уникальный код JavaScript, WEBO Site SpeedUp вынужден создавать новый файл в кэше для каждой страницы, если опция *Объединение JavaScript-файлов* включена. Если при этом каждый объединенный файл подвергается минимизации, нагрузка становится еще больше. В этой ситуации, где находится проблемный код, вы можете, например, исключить некоторые файлы из объединения или выбрать опцию *_Объединять только JavaScript, подключенный в тэге head_* или исключить из объединения встроенный код.

Другой похожий пример — когда на каждой странице содержится уникальный набор изображений, и включена опция *Объединять HTML-изображения*.

Ниже приведен список опций, которые могут приводить к подобным проблемам.

Наиболее сложные методы оптимизации WEBO Site SpeedUp, потребляющие большое число серверных ресурсов:

- (Производительность) Не проверять время изменения файлов (если отключено)
- (Объединение CSS) Объединять CSS, подключенный в тэгах `<head>` and `<body>`
- (Объединение JavaScript) Объединять JavaScript, подключенный в тэгах `<head>` и `<body>`
- (Минимизация) Минимизировать с помощью YUI Compressor
- (Минимизация) Сжать HTML до 1 строки

Другие методы оптимизации WEBO Site SpeedUp, потребляющие значительное число серверных ресурсов:

- (Производительность) Не применять регулярные выражения (если отключено)
- (Производительность) Восстанавливать CSS-свойства
- (Ненавязчивый JavaScript) Переместить весь JavaScript-код перед `</body>`
- (Ненавязчивый JavaScript) Перенести вызовы JavaScript-информеров перед `</body>`
- (Ненавязчивый JavaScript) Перенести рекламные вызовы JavaScript перед `</body>`
- (Ненавязчивый JavaScript) Перенести вызовы JavaScript-счетчиков перед

</body>

- (Ненавязчивый JavaScript) Отложить загрузку фреймов
- (CSS Sprites) Применять CSS Sprites
- (CSS Sprites) Объединять HTML-изображения

Также если на страницах сайта находится большое число изображений, файлов CSS, JavaScript (порядка ста и более), то с каждым просмотром страницы может возникать значительная нагрузка на сервер, если включены следующие опции:

- (CDN) Распределять изображения
- (CDN) Распределять CSS-файлы
- (CDN) Распределять JavaScript-файлы

Если вам не удалось найти опции WEBO Site SpeedUp, которые снижают производительности вашего сервера, попробуйте создать новую конфигурацию и подключать опции одну за другой, чтобы определить искомое слабое место.

Все еще не нашли решения? Попробуйте отключить WEBO Site SpeedUp и снова проверьте производительность сервера, чтобы убедиться, что проблема не вызвана каким-либо другим приложением или подключенным расширениями.

Отсутствие поддержки множественных хостов

Если требуется включить несколько параллельных хостов для ускорения загрузки статических ресурсов, то стоит проверить следующие моменты:

- Наличие поддержки этих хостов в DNS. Для этого нужно сделать соответствующие этим хостам записи в вашей DNS-зоне, указывающие на требуемый IP-адрес (обычно тот же, что и у текущего сайта).
- Включить поддержку этих хостов на уровне самого сервера. Для Apache это делается директивой `ServerAlias`, например:

```
ServerAlias i1.site.com  
ServerAlias i2.site.com
```

- Проверить, что эти хосты зеркалируют основной сайт. Для этого нужно взять адрес любого статического объекта на сайте (например, `site.com/images/my.png`) и попробовать его открыть через все дополнительные хосты (`i1.site.com/images/my.png`). При наличии каких-либо проблем необходимо повторить предыдущие шаги.
- Добавить указанные хосты в конфигурацию WEBO Site SpeedUp

```
Множественные хосты -> Доступные хосты -> Название хостов через  
пробел
```

- и включить поддержку множественных хостов

```
Множественные хосты -> Включить параллельные хосты, например, i1  
i2 -> Да
```

WEBO Site SpeedUp автоматически проверяет несколько наиболее распространенных хостов и последняя настройка может и не потребоваться.

После описанных действий все изображения на сайте будут отдаваться через несколько хостов, что существенно ускорит загрузку каждой страницы.

Множественные хосты не сохраняются

WEBO Site SpeedUp автоматически проверяет все введенные хосты на доступность, чтобы быть уверенным, что их можно использовать для параллельных загрузок. Вы можете отключить данную проверку (если в силу каких-либо причин она производится некорректно):

```
Множественные хосты -> Автоматически проверять доступность хостов  
-> Нет
```

Ресурсы сервера ограничены

В этом случае можно использовать цепочную оптимизацию:

- Загружаем главную страницу сайта (в виде простого HTML) в кэширующую директорию и сохраняем в файле compressing.php.
- Вставляем в файл compressing.php вызовы WEBO Site SpeedUp.
- Перенаправляем пользователя на страницу compressing.php со специфическим GET-параметром (web_optimizer_stage).
- Файл compressing.php перенаправляет сам на себя перед созданием CSS Sprites, после п.16 в логике создания CSS Sprites (repeat-x, repeat-y, no-repeatb, no-repeatr), после создания всех CSS Sprites и после включения data:URI в файл стилей (всего 5 редиректов).
- Удаляем файл compressing.php.
- Обновляем текущий index.php (при этом все необходимые файлы в кэше для отображения главной страницы уже присутствуют).

5.4. Техническая поддержка

Пожалуйста попробуйте решить вашу проблему при помощи раздела устранение неполадок.

Если ваша проблема не описана в этом разделе попробуйте [найти вашу проблему](#) в [списке задач проекта](#). Скорее всего ваша проблема уже известна и будет решена в одном из ближайших обновлений. Если вы не нашли описание вашей проблемы и там, сообщите нам о ней при помощи [формы создания новой задачи](#) (необходима учетная запись Google). Все важные проблемы обычно решаются в течение суток (часто в течение нескольких часов). Пожалуйста, создавайте отдельные задачи по каждой отдельной проблеме.

Платные пользователи WEBO Site SpeedUp обладают приоритетной поддержкой в режиме 24/7, и они могут решить любые проблемы с WEBO Site SpeedUp в интерактивном режиме, используя нашу [контактную информацию](#).