

# Переводим Magento 1 на PHP7, часть 2

Итак, в [прошлой статье](#) мы перевели нашу (вашу) Magento на PHP7. Увы, весьма часто возникают ситуации, когда несмотря на установку патчей и плагинов совместимости, под свежим PHP, Мадженто работает некорректно. Давайте же рассмотрим основные риски и способы их устранения.

1. Устаревшая версия Magento. За последние 2 года система развивалась очень бурно. Разработчики практически несколько раз в год выпускали новые версии. Поблагодарить за это, в том числе, стоит и хакеров, которые внезапно очень пристально обратили на Magento свой взор. Но тем и лучше, т.к. ни одна другая система электронной коммерции не получала настолько подробного стресс-тестирования. Тем не менее, минусом данной ситуации стоит считать то, что обладатель или админ старенькой версии Magento, вроде 1.4 или 1.5 вынужден будет поставить большое количество патчей, чтобы закрыть основные дыры в безопасности. А ведь патчи могут быть “капризными”, и если ваша система в чем-то была модернизирована (поправили какие-то системные файлы, не создавая копии в local), то “вычесывание блох” и натягивание патчей может занять много времени. Так что, оптимальным вариантом для перехода на PHP7 следует считать версию Magento от 1.9, а еще лучше самую последнюю на текущий момент (речь идет только о семействе 1-й версии, разумеется). Итак: владельцам старых версий Magento проще будет проапгрейдить свою систему до актуального варианта, чем лепить заплатки на устаревшую версию.

---

2. Устаревшие плагины. Увы, в отличие от того же WordPress, Magento не умеет автоматически обновлять плагины. В лучшем случае, плагин предложит вам обновиться, показав всплывающее окно при входе в админку, и оставит напоминание на страничке системных новостей. Однако, подавляющая часть плагинов такого функционала не имеет. Так что, если вы испытываете какие-либо проблемы с некорректной работой

системы после перехода на PHP7, то для начала отключите все сторонние (не-Mage) плагины, и проверьте, не ушла ли ошибка. Отключать плагины лучше посредством замены ключа `<active>true</active>` на `<active>>false</active>` в настроечном файле плагина. XML-файлы настроек находятся в папке `app/etc/modules`. И не забудьте перед правкой отключить компиляцию! Также, имейте в виду, что плагины на самом деле не выключаются при выборе опции “отключить вывод плагина” в разделе конфигурации системы (*система > конфигурация > расширенные*). После отключения нештатных плагинов, можно по одному включать их, и проверять работу системы. Так же, нелишним будет обновить плагины до последних версий.

---

3. Неполная правка системных файлов. Для корректной работы Magento с PHP7, увы, недостаточно только поправить файл `Layout.php`, например, при помощи патча из [варианта №1 прошлой статьи](#). После правки `Layout.php` система, разве что, начинает запускаться и более-менее полноценно работать. Но, также возможны и проблемы, рассмотрим их ниже:

Внимание: при необходимости правки системного файла (из какой-либо подпапки `app/code/core/Mage`), обязательно создавайте local-копию этого файла (`app/code/local/Mage/полный/путь/источника`), и все правки вносите только в копию. Если вы поправите системный файл прямо в папке `core`, то в дальнейшем будет затруднена установка новых системных патчей.

- Не работает экспорт товара. Правим файл:

```
app\code\core\Mage\ImportExport\Model\Export\Entity\Product\Type\nAbstract.php
```

```
перенеся его в  
app\code\local\Mage\ImportExport\Model\Export\Entity\Product\Typ\nAbstract.php
```

Строчку:

```
$data['filter_options'] = $this->$data['options_method']();
```

заменяем на:

```
$data['filter_options'] = $this->{$data['options_method']}();
```

- Не работает экспорт клиентов. Правим:

```
app\code\core\Mage\ImportExport\Model\Export\Entity\Customer.php
```

Строчку:

```
$data['filter_options'] = $this->$data['options_method']();
```

заменяем на:

```
$data['filter_options'] = $this->{$data['options_method']}();
```

- Не работает загрузчик файлов в админке. Правим:

```
lib\Varien\File\Uploader.php -- его можно поправить прямо по месту
```

Строчку:

```
$params['object']->$params['method']($this->_file['tmp_name']);
```

заменяем на:

```
$params['object']->{$params['method']}($this->_file['tmp_name'])  
;
```

- Покупатели не могут войти в систему. Правим:

```
app\code\core\Mage\Core\Model\Resource\Session.php  
строка 218 (или рядом где-то, зависит от версии)
```

Строчку:

```
return $data;
```

заменяем на:

```
return (string)$data;
```

- Некорректно считается общая сумма заказа. Скачиваем и устанавливаем [плагин](#).

После всех произведенных правок, обязательно производим перекомпиляцию (если включена компиляция) и обновляем кэш системы.

---

# Переводим Magento 1 на PHP7, часть 1

Язык PHP постоянно развивается, и в настоящее время актуальной является уже 7-я его версия. Интерпретатор этой версии работает заметно быстрее предыдущих – многие веб-мастера отмечают 2-3 кратный прирост производительности скриптов, и, соответственно, весьма заметное понижение нагрузки на сервер.

Magento – это довольно “тяжелая” система, и перевод на PHP7 для неё весьма желателен. Но, вплоть до Magento 2, система была рассчитана на работу под PHP 5.x, и с 7-ми версиями PHP “из коробки” просто так не работает.

К счастью, несовместимость эту можно исправить довольно легко, не прибегая к полной переустановке или апгрейду системы (которые, как любой переезд, иной раз хуже пожара).

Перейдем к сути. Во-первых, перед заменой версии PHP, в Magento необходимо отключить кэш и компиляцию. И, да: вы же не производите апгрейд PHP непосредственно на рабочем сервере? Если по каким-то причинам все же производите, то создайте в корневом каталоге сайта файл *maintenance.flag*, он переведет Magento в режим обслуживания. Разумеется, после полного бэкапа файлов и баз(ы) данных.

Далее, выбираем один из вариантов приведения Magento в PHP7-совместимое состояние:

Вариант 1. С минимальным вмешательством, подходит для более-менее свежих версий Magento CE, вроде 1.9.2.4. Патчим систему:

```
# запускаем в корневом каталоге Magento

# одной строкой, с использованием wget
wget -q0-
'https://gist.githubusercontent.com/MageMechanic/66449504110fbbd
96214/raw/php7_magento_1_9_core.patch' | patch -p0

# одной строкой, с использованием curl
curl -s
'https://gist.githubusercontent.com/MageMechanic/66449504110fbbd
96214/raw/php7_magento_1_9_core.patch' | patch -p0

# более наглядно:
PATCH_URL='https://gist.githubusercontent.com/MageMechanic/66449
504110fbbd96214/raw/php7_magento_1_9_core.patch'

# применяем через wget
wget -q0- $PATCH_URL | patch -p0

# ...или через curl
curl -s $PATCH_URL | patch -p0

# откатить изменения через wget
wget -q0- $PATCH_URL | patch -p0 -R

# ...или через curl
curl -s $PATCH_URL | patch -p0 -R
```

Вариант 2. Решение от того же автора, что и патч выше. Устраняет так же потенциальные проблемы с загрузкой изображений при создании/редактировании товара. Рекомендуется для версий Magento CE от 1.9 до 1.9.3.1.

[Скачиваем](#) и ставим плагин вручную (распаковываем в корневой каталог Magento).

*или*

Устанавливаем через [modman](#):

```
modman clone https://github.com/MageMechanic/PhpSeven.git --
branch 1.0.3
```

или

Устанавливаем через [composer](#):

```
composer config repositories.magemechanic git
https://github.com/MageMechanic/PhpSeven.git
composer require "magemechanic/phpseven:1.0.3"
```

Вариант 3. Решение от старых и проверенных Magento-разработчиков, компании Inchoo. Рекомендуется для версий Magento CE от 1.9, подходит и для Magento EE от 1.4.2.2. Плагин хорош полной обратной совместимостью с PHP 5.x, т.е, в случае необходимости откатить версию PHP, вам не понадобится выключать этот плагин или отменять патч. Также, рекомендован к установке после применения патча [SUPEE-8788](#) (о котором будет отдельная статья). Устанавливается просто: [скачиваем](#) и распаковываем в коренной директории Magento.

Итак, после применения патча или установки одного из плагинов совместимости, вы получите Magento, работающую под PHP7. Поздравления! Теперь можно удалить (или переименовать) файл *maintenance.flag*, и проверить работу вашего интернет-магазина. Если в вашей системе не было устаревших сторонних плагинов, то все должно будет сразу корректно заработать, и можно будет включить кэш и скомпилировать заново исполняемые файлы (не просто включить компиляцию!). Учтите, проверке должны подвергнуться все аспекты работы магазина: от регистрации и логина клиента, совершения им разных типов покупок, до создания продукта в админке, правке категорий, атрибутов и типов продуктов. Увы, как показывает практика, после перехода на PHP7 может возникнуть немало неочевидных проблем, и о них мы поговорим в

следующей части статьи.

*Источники:*

<https://gist.github.com/MageMechanic>

<http://inchoo.net/magento/its-alive/>