

## Простой вариант редиректа (.htaccess или httpd.conf)

Перенаправление всего сайта (включая подкаталоги) на новый сайт или страницу (не забывайте поставить последний "/", если переадресация идет на сайт).

```
Redirect 301 / http://www.domain.ru/newIndex.html
```

где:

**Redirect 301** - это инструкция, говорящая что страница перемещена / - означает, что переадресовываться будет все, с верхнего уровня сайта, включая подкаталоги, будет переадресовано на <http://www.domain.ru/newIndex.html> - новая страница или <http://www.domain.ru/> сайт (последний "/" обязателен).

Чтобы переадресовать только страницу, сохранив PR старой страницы:

```
Redirect 301 /old/veryOld.html http://www.domain.ru/new/veryNew.html
```

где:

**/old/veryOld.html** - путь и имя старой страницы

**<http://www.domain.ru/new/veryNew.html>** - адрес новой страницы.

Аналогичный синтаксис для переадресации сайта:

```
RedirectPermanent / http://www.domain.ru/
```

Пример переадресации каталога:

```
RedirectPermanent /old-directory http://www.domain.ru/new-directory/
```

Например, зашедших по <http://www.domain.ru/test> (директория test) переадресуем на [demo.domain.ru](http://demo.domain.ru), остальных на [login.domain.ru](http://login.domain.ru) (порядок следования записей важен):

```
Redirect permanent /test http://demo.domain.ru/
```

```
Redirect permanent / http://login.domain.ru/
```

## Использование mod\_rewrite (.htaccess)

Перенаправление запросов на сайт с www и без него, решается так:

```
Options +FollowSymLinks
RewriteEngine on
RewriteCond %{HTTP_HOST} ^domain.ru
RewriteRule ^(.*)$ http://www.domain.ru/$1 [R=permanent,L]
```

(!) В последней строке в месте шаблона обязательно использовать скобки (выражение "`^(.*)$`" или "`(.*)`"), иначе \$1 не заработает и при обращении на сайт будет осуществляться перенаправление только на главную страницу, то есть строка запроса будет обрезаться.

или альтернативный синтаксис:

```
Options +FollowSymLinks
RewriteEngine On
RewriteCond %{HTTP_HOST} ^domain.ru$ [NC]
RewriteRule ^(.*)$ http://www.domain.com/$1 [R=301,L]
```

Обозначение `[R=301,L]` означает: перенаправить клиента и отправить ему код статуса 301 (`R=301`) и сделать это правило последним (`L`).

Редирект старого домена на новый:

```
Options +FollowSymLinks
RewriteEngine on
RewriteRule (.* ) http://www.newDomain.ru/$1 [R=301,L]
```

Например, если необходимо чтобы вместо `rewrite.html` загружался файл `rewrite.xhtml`, добавьте в `.htaccess`:

```
RewriteEngine on
RewriteBase /
RewriteRule ^rewrite.html$ rewrite.xhtml [R=permanent]
```

Для замены всех `.html` файлов `.xhtml` файлами:

```
RewriteEngine on
RewriteBase /
RewriteRule ^(.*)\.html$ $1.xhtml [R=permanent] Редирект на PHP
<?php
header("HTTP/1.1 301 Moved Permanently");
header("Location: http://www.newDomain.ru/new/veryNew.html");
```

```
exit();  
?>
```

Лучше указать HTTP/1.1, так как более старые не поддерживают виртуальный хостинг. Не забудьте, что до вызова header, ничего не должно выводиться (например, echo или print). Поэтому этот код лучше ставить в начало php-скрипта.

Более полный вариант php редиректа с сохранением передаваемой страницы и параметров вызова:

```
<?  
$ref=$_SERVER['QUERY_STRING'];  
if ($ref!="") $ref='?'.$ref;  
header('HTTP/1.1 301 Moved Permanently');  
header('Location: http://newDomain.ru/'.$ref);  
exit();  
?>
```

### **Редирект на ASP**

```
<%@ Language=VBScript %>  
<%  
Response.Status="301 Moved Permanently"  
Response.AddHeader "Location", "http://www.newDomain.ru"  
response.end
```

### **%> Редирект на ASP.NET**

```
<script runat="server">  
private void Page_Load(object sender, System.EventArgs e)  
{  
    Response.Status = "301 Moved Permanently";  
    Response.AddHeader("Location","http://www.newDomain.ru");  
}
```

### **</script> Редирект на ColdFusion**

```
<.cfheader statuscode="301" statustext="Moved permanently">  
<.cfheader name="Location" value="http://www.newDomain.ru"> Редирект с
```

### **помощью meta refresh**

```
<meta http-equiv='refresh' content='0; url=http://newDomain.ru'>
```

где 0 - задержка переадресации в секундах, newDomain.ru - страница, куда переадресуем. Некоторые старые браузеры не поддерживают meta refresh со значением 0, для совместимости можно установить ненулевое значение, хотя, на мой взгляд это уже не актуально. Такой редирект не сможет склеить ваши сайты (с www и без) и передать PR, так как игнорируется поисковыми системами. Он возвращает код 200 OK, что соответствует обычной странице. Эта техника популярна у спамеров, поэтому ее

стоит применять только для страниц, которые не будут индексироваться.

### Редирект с помощью JavaScript

Варианты переадресации на JavaScript чаще реализуются с использованием функции `setTimeout('функция', задержка)`.

Например, автоматически сделать Click на кнопке "Submit" формы "searchform" через 0.1 сек после загрузки кода:

```
setTimeout('document.forms["searchform"].Submit.click()', 100);
```

На кнопку "Submit" можно повесить любое действие, например, открыть новый url в этом окне. Кстати такие редиректы чаще встречаются при организации Дорвеев (DorWay) - браузер Пользователя будет переадресован на другую страницу, а поисковый робот, который "не понимает" JavaScript, будет индексировать эту страницу, недоступную пользователю. На ней дорвейщики размещают текст, напичканный "нужными" ключевыми словами.

Чтобы просто переадресовать на другую страницу можно вставить после один из вариантов кода на JavaScript:

- `location="http://www.newDomain.ru";`
- `document.location.href="http://www.newDomain.ru";`
- `window.location.reload("http://www.newDomain.ru");`
- `document.location.replace("http://www.newDomain.ru");`

В последнем случае уже нельзя будет вернуться на страницу выполнившую переадресацию, так как ее адрес стирается из history, что нередко и нужно. Если нужна задержка по времени, можно оформить `location="http://www.newDomain.ru";` в виде функции и вставить ее в `setTimeout('функция()', задержка_в_мсек)`; Редирект на JavaScript не является 301 редиректом и не передаст PR страницы, не сможет обеспечить ее склейку.

### Некоторые особенности редиректов

- Методы редиректа с `.htaccess` работают только на Linux серверах, имеющих

Apache с включенным модулем Mod-Rewrite.

- Использование .htaccess создает дополнительную нагрузку на сервер Apache, более эффективно прописывать те же команды в его конфигурационном файле httpd.conf, но, как правило, к нему нет доступа у вебмастера.
- 301 редирект, позволяет сберечь трафик и передать PR страницы для поисковых систем (для Google точно).
- Процесс склейки и передачи PR занимает длительное времени - до нескольких месяцев и также зависит от поисковой системы, поэтому не удаляйте старую страницу или сайт, пока не произойдет окончательный перенос.
- Некоторые поисковые системы требуют для склейки сайтов дополнительных настроек, например, для Яндекса нужно дополнительно прописывать robots.txt

## Заключение

Безопасный способ редиректа старых страниц на новые или старого сайта на новый адрес, с сохранением позиций в поисковых системах, заключается в использовании **301 редиректа**, который также позволит вам передать старый Page Rank страницы на новый сайт.

Более подробно про mod\_rewrite можно прочитать на:

- <http://httpd.apache.org/docs/2.0/misc/rewriteguide.html>
- [http://www.egoroff.spb.ru/portfolio/apache/mod\\_rewrite.html](http://www.egoroff.spb.ru/portfolio/apache/mod_rewrite.html) (русский перевод описания модуля mod\_rewrite)

Проверить корректность редиректа и других возвращаемых заголовков, можно, например, здесь:

<http://www.rexswain.com/httpview.html>

```
{include_content_item 43}
```

где:

/old/veryOld.html - путь и имя старой страницы

<http://www.domain.ru/new/veryNew.html> - адрес новой страницы.