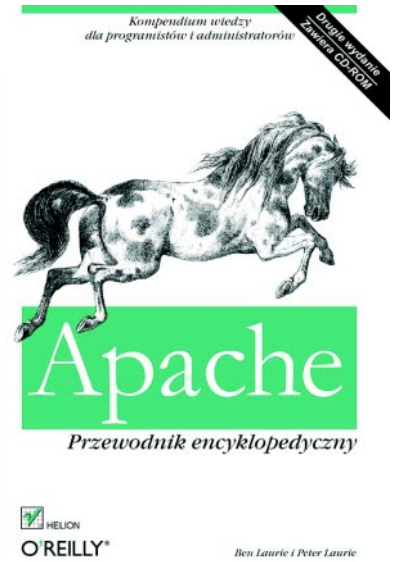
Apache przewodnik encyklopedyczny

Autorzy: Ben Laurie, Peter Laurie
Tłumaczenie: Tomasz M. Sadowski
ISBN: 83-7197-181-8
Format: B5, 408 stron
Zawiera CD-ROM

Przesyłka gratis! Odbiorca pokrywa jedynie koszty pobrania (2,70 zł)
w przypadku przesyłki za zaliczeniem pocztowym



Wydawnictwo Helion
ul. Chopina 6, 44-100 Gliwice, POLAND
telefon: (32) 230-98-63, 231-22-19
fax: (32) 230-98-63 w.10
mail: helion@helion.com.pl

Książka ta stanowi kompendium wiedzy o najpopularniejszym serwerze WWW na świecie. Przedstawione w niej przykłady umożliwią świeżo upieczonym administratorom serwerów WWW błyskawiczne nabranie wprawy w zarządzaniu serwerem. Doświadczeni administratorzy i programiści tworzący skrypty CGI znajdą tu wiele cennych, szczegółowych informacji natury encyklopedycznej, zaś dla programistów systemowych przeznaczone są rozdziały o interfejsie programowym i tworzeniu modułów.

- Zobacz przykładowy rozdział
- Spis treści
- Jeżeli znasz tę książkę oceń ją
- Aktualny cennik książek e-mailem
- Książki i "3D" Online
- Informacje o nowościach e-mailem
- Zamów najnowszy katalog

© Helion 2000

8

Przeadresowywanie

Mało co w życiu znajduje się na właściwym miejscu we właściwym czasie. Ten smutny wniosek dotyczy nie tylko serwerów WWW, ale i wszelkich innych rzeczy, które nas otaczają. Aby poradzić sobie z tym problemem, administrator ma do dyspozycji dyrektywy `Alias` i `Redirect`, pozwalające kierować nadchodzące żądania na inne tory, bądź to w obrębie systemu plików serwera, bądź w inne miejsca sieci. Co prawda w idealnych warunkach nie powinno to być konieczne, jednak w praktyce możliwość przenoszenia dokumentów do innych katalogów – a nawet na inne serwery – bez konieczności poprawiania całej struktury łącz jest często bardzo pożądana[1]. Bardziej uzasadnionym praktycznie zastosowaniem, przynajmniej jeśli idzie o dyrektywę `Alias`, jest organizowanie struktury katalogów, rozrzuconych nieraz po całym systemie. Katalogi takie mogą być zarządzane przez różnych użytkowników, a w skrajnych przypadkach mogą nawet znajdować się w systemach plików montowanych jako zdalne (sieciowe). W takiej sytuacji dyrektywa `Alias` pozwala zaprowadzić w ich strukturze pewien porządek.

Dyrektywa `ScriptAlias` umożliwia uruchamianie skryptów CGI, bez których niewiele witryn byłoby w stanie funkcjonować. Alternatywą, udostępniającą wszystkie możliwości dyrektywy `ScriptAlias` i sporo więcej, jest nowa dyrektywa `Rewrite`, którą omówimy w dalszej części rozdziału. Użycie tej ostatniej wymaga niestety włożenia pewnego wysiłku w zaprogramowanie kilku rzeczy.

Dyrektywa `ScriptAlias` jest stosunkowo prosta w użyciu, ale z drugiej strony stanowi przykład „modułowej-ale-nie-do-końca" struktury serwera Apache. Chociaż `ScriptAlias` jest zdefiniowana w module *mod_alias.c*, implementujący ją kod odwołuje się do instrukcji zawartych w module *mod_cgi.c* (ewentualnie innego modułu obsługującego interfejs CGI). Funkcje zawarte w module *mod_alias* zapewniają jeden z mechanizmów obsługujących działanie skryptów CGI, zaś sam moduł włączany jest do kodu serwera automatycznie podczas kompilacji.

Plik konfiguracji serwera zawarty w katalogu *…site/alias* ma następującą postać:

```
User webuser
Group webgroup
ServerName www.butterthlies.com.pl

NameVirtualHost 192.168.123.2

<VirtualHost www.butterthlies.com.pl>
```

[1] Pamiętajmy, że zbyt wiele takich manewrów potrafi skutecznie utrudnić zarządzanie witryną.

```
ServerAdmin sales@butterthlies.com.pl
DocumentRoot /usr/www/site.alias/htdocs/customers
ErrorLog /usr/www/site.alias/logs/customers/error_log
TransferLog /usr/www/site.alias/logs/customers/access_log
Alias /gdzieindziej /usr/www/somewhere_else/
</VirtualHost>

<VirtualHost sales.butterthlies.com.pl>
ServerAdmin sales_mgr@butterthlies.com.pl
DocumentRoot /usr/www/site.alias/htdocs/salesmen
ServerName sales.butterthlies.com.pl
ErrorLog /usr/www/site.alias/logs/salesmen/error_log
TransferLog /usr/www/site.alias/logs/salesmen/access_log
Redirect /sekrety http://www.butterthlies.com.pl
ScriptAlias /cgi-bin /usr/www/cgi-bin
</VirtualHost>
```

ScriptAlias

```
ScriptAlias prefiks-URL katalog
U¿ycie: konfiguracja g³ówna, serwery wirtualne
```

Z dyrektywą `ScriptAlias` mieliśmy już do czynienia w rozdziale 4. Umożliwia ona umieszczenie skryptów w katalogu położonym z dala od wścibskich oczu i palców, a ponadto automatycznie oznacza tenże katalog jako zawierający skrypty CGI.

ScriptAliasMatch

```
ScriptAliasMatch wyra¿enie-regularne katalog-lub-plik
U¿ycie: konfiguracja g³ówna, serwery wirtualne
```

Dyrektywa ta działa podobnie jak `ScriptAlias`, z tą różnicą, że odebrany adres URL jest porównywany z wyrażeniem regularnym. W przypadku stwierdzenia zgodności, fragmenty adresu odpowiadające zawartości nawiasów są podstawiane do nazwy *katalogu*, a uzyskany w ten sposób łańcuch traktowany jest jak nazwa pliku. Aby np. skierować odwołanie do standardowego katalogu */cgi-bin*, możemy zapisać:

```
ScriptAliasMatch ^/cgi-bin/(.*) /usr/local/apache/cgi-bin/$1
```

Alias

```
Alias prefiks-URL katalog
U¿ycie: konfiguracja g³ówna, serwery wirtualne
```

Dyrektywa `Alias` pozwala na przechowywanie dokumentów w miejscu innym, niż główny katalog dokumentów określony dyrektywą `DocumentRoot`. W ramach przykładu możemy utworzyć katalog o nazwie */usr/www/somewhere_else* i umieścić w nim plik *lost.txt*, zawierający prosty komunikat:

```
Zgubi³em siê
```

Teraz należy poddać edycji plik *httpd.conf*, uzyskując w efekcie coś takiego:

```
...
TransferLog /usr/www/site.alias/logs/customers/access_log
Alias /gdzieindziej /usr/www/somewhere_else/
</VirtualHost>
...
```

Po uruchomieniu serwera poleceniem `./go`, odwołanie do adresu *http://www.butterthlies.com.pl/ gdzieindziej* powinno dać następujący wynik:

```
Index of /somewhere_else
* Parent Directory
* lost.txt
```

Kliknięcie łącza `Parent Directory` spowoduje powrót do głównego katalogu dokumentów (`DocumentRoot`) witryny, czyli */usr/www/site.alias/htdocs/customers*, a nie, jak można byłoby przypuszczać, katalogu */usr/www*. Dzieje się tak dlatego, iż łącze `Parent Directory` oznacza nie katalog nadrzędny, lecz „nadrzędny adres URL", czyli w naszym przypadku *http://www.butterthlies.com.pl*.

Nieco zaskakujący (nawet dla tych, którzy już wiedzą, ale akurat sobie zapomnieli) jest fakt, że jeśli w głównym katalogu dokumentów witryny *http://www.butterthlies.com.pl* nie umieścimy pliku *index.html*, automatycznie wygenerowany indeks nie będzie zawierał katalogu *gdzie_indziej*.

Pamiętaj również, aby przypadkiem nie użyć formy

```
Alias /gdzieindziej/ /usr/www/somewhere_else
```

(pierwszy parametr zakończony jest ukośnikiem). Wynikiem takiego zapisu mogą być nieco konfundujące komunikaty o nieznalezieniu dokumentu („Not Found").

AliasMatch

```
AliasMatch wyra¿enie-regularne katalog
U¿ycie: konfiguracja g³ówna, serwery wirtualne
```

Dyrektywa ta jest analogiczna do `Alias`. Jedyną różnicą jest pierwszy parametr, dany nie konkretnym fragmentem nazwy, a wyrażeniem regularnym.

UserDir

```
UserDir katalog
Wartoœæ domyœlna: public_html
U¿ycie: konfiguracja g³ówna, serwery wirtualne
```

Działanie dyrektywy `UserDir` omówimy na przykładzie. Załóżmy, że klient żąda dostępu do danych położonych w katalogu macierzystym (ang. *home directory*) użytkownika *pawel*. Użyty w żądaniu adres URL ma w takiej sytuacji postać *http://www.butterthlies.com.pl/ ~pawel*, co tłumaczy się jako „katalog macierzysty Pawła na serwerze identyfikowanym w systemie DNS jako *www.butterthlies.com.pl*". Dyrektywa `UserDir` pozwala zdefiniować faktyczny podkatalog katalogu macierzystego użytkownika, używany podczas obsługi żądania dostępu do dokumentu podanego w formacie przedstawionym powyżej. Parametr *katalog* może zawierać:

- nazwę lub wzorzec nazwy katalogu, jak podano w przykładach poniżej;
- słowo kluczowe `disabled`. Jego użycie blokuje odwzorowywanie nazw użytkowników na katalogi; nie dotyczy to użytkowników wyszczególnionych na liście `enabled`;
- słowo kluczowe `disabled` i listę nazw użytkowników rozdzielonych znakami spacji. Lista ta określa nazwy użytkowników, dla których odwzorowywanie nie będzie wykonywane *nawet* w przypadku ich umieszczenia na liście `enabled`;

- słowo kluczowe `enabled` i listę nazw użytkowników rozdzielonych znakami spacji. Lista ta określa nazwy użytkowników, dla których odwzorowywanie będzie dokonywane mimo globalnego zakazu ustanowionego słowem `disabled`.

Jeżeli dyrektywa `UserDir` nie zawiera słowa `disabled` ani `enabled`, jej parametr interpretowany jest jako wzorzec nazwy pliku używany do tłumaczenia nazwy użytkownika na specyfikację katalogu. Oto kilka przykładów tłumaczenia adresu URL: *http://www.firma.com/ ~pawel/raz/dwa.html*:

```
UserDir public_html      -> ~pawel/public_html/raz/dwa.html
UserDir /usr/web         -> /usr/web/pawel/raz/dwa.html
UserDir /home/*/www      -> /home/pawel/www/raz/dwa.html
```

Z kolei poniższe użycie poniższych dyrektyw spowoduje przesłanie do klienta polecenia przeadresowania (*redirect*):

```
UserDir http://www.firma.com/biuro   ->
http://www.firma.com/biuro/pawel/raz/dwa.html
UserDir http://www.firma.com/*/usr   ->
http://www.firma.com/pawel/usr/raz/dwa.html
UserDir http://www.firma.com/~*      ->
http://www.firma.com/~pawel/raz/dwa.html
```

Użycie dyrektywy `UserDir` wymaga jednak rozwagi: przykładowo, zapis `UserDir ./` spowoduje odwzorowanie na adres „/” katalogu */~root*, czego raczej należy unikać. Użytkownikom serwera Apache w wersji 1.3 lub nowszej usilnie zaleca się umieszczenie w pliku konfiguracji serwera dyrektywy `UserDir disabled root`.

WIN32 Wersja Apache przeznaczona dla Windows nie zna pojęcia katalogu macierzystego użytkownika, toteż odwzorowania kierujące odwołanie do takiego katalogu (zobacz pierwszy przykład) nie będą działały poprawnie.

Redirect

```
Redirect prefiks-URL URL
U¿ycie: konfiguracja g³ówna, serwery wirtualne, katalogi, pliki
.htaccess
```

Dyrektywa `Redirect` kieruje odwołanie do adresu URL pod inny adres.

RedirectMatch

```
RedirectMatch wyra¿enie-regularne URL
U¿ycie: konfiguracja g³ówna, serwery wirtualne
```

Działanie dyrektywy `RedirectMatch` jest analogiczne do `Redirect`, jednak pierwszy parametr jest w tym przypadku dany wyrażeniem regularnym.

Z niejaką przykrością musimy stwierdzić, że pracownicy działu handlowego naszej firmy od pewnego czasu nadużywali swoich przywilejów i posiadanych zasobów. W związku z tym przełożeni postanowili dać im nauczkę, zabraniając im dostępu do różnych słodkich sekretów zapisanych w pliku *sekrety* i odsyłając ich w zamian do publicznie dostępnej strony przeznaczonej dla klientów. Cóż za upokorzenie! Ale zrobić to łatwo: wystarczy poprawić plik *httpd.conf* do następującej postaci:

```
...
<VirtualHost sales.butterthlies.com.pl>
ServerAdmin sales_mgr@butterthlies.com.pl
Redirect /sekrety http://www.butterthlies.com.pl
```

```
DocumentRoot /usr/www/site.alias/htdocs/salesmen
...
```

Dyrektywa `Redirect` może być położona w dowolnym miejscu w obrębie odpowiedniego bloku `<VirtualHost>`. Jeśli teraz spróbujesz odwołać się do adresu URL *http://sales.butterthlies.com.pl/sekrety*, zostaniesz skierowany „objazdem" na stronę przeznaczoną dla klientów – *http://www.butterthlies.com.pl*.

Bardzo istotną różnicą pomiędzy dyrektywami `Alias` i `Redirect` jest to, że w przypadku `Redirect` przeglądarka otrzymuje jawną informację o nowym adresie, zaś w przypadku dyrektywy `Alias` – nie. Adres przekazany klientowi w wyniku użycia dyrektywy `Redirect` jest traktowany jako adres bazowy dla wszystkich łącz znajdujących się w pobranym dokumencie.

Translacja adresów URL

W pierwszej części rozdziału zajęliśmy się modułem *mod_alias.c* i jego przyległościami. Zestaw funkcji modułu *alias*, a także wiele innych możliwości, implementuje inny moduł, *mod_rewrite.c*. Jest to zwarty zbiór procedur, będący sam w sobie niemalże kompletnym programem[2]. Dokładna i obszerna dokumentacja modułu dostępna jest pod adresem *http://www.engelschall.com/pw/apache/rewriteguide/*, do którego odsyłamy Czytelników poważniej zainteresowanych wykorzystaniem jego możliwości. Tutaj ograniczymy się jedynie do ogólnego omówienia możliwości i funkcji modułu.

Dyrektywa `Rewrite` porównuje adres URL z tzw. *wzorcem translacji* (ang. *rewriting pattern*). W przypadku stwierdzenia zgodności wzorca i adresu, ten ostatni jest następnie tłumaczony (przepisywany) poprzez *podstawienie translacyjne* (ang. *rewriting substitution*). Wzorzec translacji dany jest zwykle prostym wyrażeniem regularnym, jak np. zapis `mod.*\.c`, odpowiadający dowolnej nazwie modułu. Teoria i praktyka użycia wyrażeń regularnych jest dość obszerna, toteż zainteresowanych Czytelników odsyłamy do dokumentu *…/src/regex/regex.7* (jest to strona podręcznika systemowego *man*, którą np. w systemie FreeBSD można wyświetlić poleceniem `nroff -man regex.7`). Wyrażenia regularne omówiono również w dokumentacji standardu POSIX, a także w książce Jeffreya Friedla *Mastering Regular Expressions* (O'Reilly & Associates). W ogromnym skrócie można powiedzieć, że wyrażenia regularne pozwalają na tworzenie wzorców, zbudowanych z określonego zestawu znaków specjalnych i wykorzystywanych do porównywania z fragmentami przekazywanych przez klienta adresów URL.

W podstawieniach można wykorzystywać funkcje odwzorowujące, pobierające fragmenty adresów i porównujące je z zawartością baz danych, a nawet przekształcające je za pomocą innych programów. Proces tłumaczenia adresu może być prowadzony iteracyjnie lub rekurencyjnie; według dokumentacji możliwe jest tworzenie „pętli tłumaczących, definiowanie kryteriów stopu, sekwencyjne użycie reguł, tworzenie konstrukcji pseudowarunkowych, wymuszanie przeadresowań, typów MIME i przekazywania poprzez moduł serwera pośredniczącego". Zestaw funkcji oferowanych przez moduł *mod_rewrite* jest tak obszerny, iż jego opanowanie na drodze teoretycznej jest najprawdopodobniej niemożliwe. Wydaje się wręcz, iż żaden z problemów związanych z omawianym przez nas tematem nie jest dla *mod_rewrite* nierozwiązywalny,

[2] Nie zmienia to faktu, że proste zadania łatwiej jest realizować w oparciu o dyrektywę `Alias` i jej dodatki.

zakładając oczywiście, że programista będzie w stanie włożyć w rozwiązanie odpowiedni wysiłek intelektualny.

Zastosowania modułu *mod_rewrite* sprowadzają się do czterech dziedzin.

- Użycia go przez administratora serwera w pliku *httpd.conf* w kontekście głównym. Zdefiniowane tak reguły mają zastosowanie do wszystkich adresów obsługiwanych przez serwer, łącznie z adresami znajdującymi się w domenie serwerów wirtualnych.
- Użycia go przez administratora serwera w blokach `<VirtualHost>`. Zdefiniowane tak reguły stosują się tylko do adresów związanych z danym serwerem wirtualnym.
- Użycia go przez administratora w blokach `<Directory>`. Tak zdefiniowane reguły odnoszą się wyłącznie do danego katalogu.
- Stosowania go przez użytkowników w ich własnych plikach *.htaccess*. Zdefiniowane tak reguły odnoszą się wyłącznie do danego katalogu.

Same dyrektywy nie są skomplikowane; omawiamy je poniżej.

RewriteEngine

```
RewriteEngine on|off
U¿ycie: konfiguracja g³ówna, serwery wirtualne, katalogi, pliki .htaccess
```

Dyrektywa `RewriteEngine` włącza (`on`) i wyłącza (`off`) mechanizm translacji adresów. Użycie parametru `off` całkowicie blokuje translację. Zamiast wyłączać poszczególne reguły poprzez wstawianie do wierszy `RewriteRule` znaków komentarza, lepiej jest wyłączyć translację całkowicie dyrektywą `RewriteEngine off`.

RewriteLog

```
RewriteLog plik
U¿ycie: konfiguracja g³ówna, serwery wirtualne
```

Dyrektywa ta określa nazwę pliku, w którym rejestrowane będą zdarzenia związane z translacją adresów URL. Nazwy nie rozpoczynające się od znaku ukośnika („/”) są traktowane jako względne w stosunku do głównego katalogu serwera (`ServerRoot`). Dyrektywa ta powinna występować w pliku konfiguracji serwera tylko raz.

RewriteLogLevel

```
RewriteLogLevel liczba
Wartoœæ domyœlna: 0
U¿ycie: konfiguracja g³ówna, serwery wirtualne
```

Dyrektywa ta określa poziom szczegółowości danych rejestrowanych w pliku dziennika. Wartość 0 oznacza brak rejestracji; wartość 9 nakazuje rejestrowanie praktycznie wszystkich zdarzeń. Uwaga: użycie wartości powyżej 2 powoduje zauważalne spowolnienie pracy serwera.

RewriteMap

```
RewriteMap mapa txt|dbm|prg|rnd|int:plik
U¿ycie: konfiguracja g³ówna, serwery wirtualne
```

Dyrektywa `RewriteMap` określa zewnętrzny *plik* zawierający *mapę podstawień*. Łańcuchy używane do podstawień są ustalane poprzez odnalezienie w pliku zadanego klucza i odczytanie odpowiadającej mu wartości. Format zapytania o wartość klucza jest następujący:

```
${mapa : klucz | wartoœæ-domyœlna}
```

przy czym *wartość-domyślna* jest ignorowana w przypadku nieznalezienia *klucza*. Typ pliku zawierającego mapę podstawień określany jest drugim parametrem dyrektywy, który może mieć wartość:

`txt`
: Mapa podstawień zapisana jest w postaci pliku tekstowego (ASCII). Plik taki może zawierać puste wiersze, komentarze (wiersze zaczynające się od znaku #) i oczywiście wiersze danych w postaci

```
wartoœæ-klucza wartoœæ-podstawiana
```

`dbm`
: Mapa podstawień zapisywana jest w pliku DBM o strukturze rozproszonej (ang. *DBM hashfile*), czyli tzw. „nowym" formacie binarnym DBM (NDBM; format ten liczy sobie obecnie już ponad 15 lat i używany jest też do przechowywania danych używanych przez moduł *auth_dbm*). Plik taki zawiera te same dane, co opisana poprzednio reprezentacja tekstowa. Do jego utworzenia można wykorzystać któreś z dostępnych w systemie narzędzi do zarządzania plikami *ndbm*, ewentualnie dostarczany w pakiecie Apache skrypt *dbmmanage*.

`prg`
: W tym przypadku „mapa" jest wywoływanym przez serwer Apache programem (plikiem wykonywalnym lub skryptem CGI). Wartość klucza przekazywana jest do standardowego wejścia programu jako łańcuch zakończony znakiem nowego wiersza, natomiast wartość podstawiana zwracana jest poprzez standardowe wyjście. W przypadku nieznalezienia klucza zwracany jest łańcuch `NULL`. Dokumentacja programu Apache zawiera trzy istotne uwagi dla programistów:

  - Program powinien być prosty, bowiem jego załamanie kończy się zawieszeniem całego serwera Apache.
  - Operacje wyjściowe wykorzystujące standardowe wyjście nie mogą być buforowane, gdyż prowadzi to do powstania impasu. Programy pisane w języku C powinny zawierać instrukcję

    ```
    setbuf(stdout,NULL);
    ```

    zaś programy w Perlu

    ```
    select(STDOUT); $|=1;
    ```

  - Wymiana danych z programem powinna być synchronizowana za pomocą pliku blokady (służy do tego dyrektywa `RewriteLock`).

`rnd`

Mapa podstawień jest przechowywana w tzw. *randomizowanym pliku tekstowym* (ang. *randomized plain text file*). Format ten jest zbliżony do zwykłego formatu tekstowego, wyposażony jest jednak w pewną dodatkową cechę: odszukana wartość klucza jest traktowana jako zestaw wartości alternatywnych rozdzielonych znakami kreski pionowej („|"). Jako wynik wyszukiwania zwracana jest losowo wybrana wartość z zestawu. Rozwiązanie takie wygląda na pozór absurdalnie, jednak ma ono swoje zastosowanie – jest nim równoważenie obciążenia w przypadku pracy Apache jako serwera odwrotnie pośredniczącego (ang. *reverse proxy*). W takiej sytuacji przeszukiwana baza zawiera nazwy serwerów, a każde żądanie przesyłane do serwera pośredniczącego jest kierowane do losowo wybranego serwera znajdującego się „za nim".

`int`

Podstawienia realizowane są przez wewnętrzną funkcję serwera Apache `toupper()` lub `tolower()`. Funkcje te dokonują konwersji przekazanej wartości na wyłącznie WIELKIE lub wyłącznie małe litery.

RewriteBase

```
RewriteBase bazowy-adres-URL
U¿ycie: katalogi, pliki .htaccess
```

Dyrektywa `RewriteBase` jawnie ustala bazowy adres URL używany w translacji adresów wykonywanej dla indywidualnych katalogów. Co prawda ten sam efekt można osiągnąć konstruując odpowiedni zestaw reguł translacji, jednak czasami prościej jest użyć rozwiązania „zbiorczego". Adresy przekazywane do reguł translacji umieszczonych w pliku *.htaccess* pozbawione są fragmentu zawierającego nazwę katalogu lokalnego, toteż przetwarzanie dotyczy wyłącznie pozostałej części adresu. Po zakończeniu przetwarzania, do wynikowego adresu dołączany jest prefiks ustalony za pomocą dyrektywy `RewriteBase`. Przykład zacytujemy za dokumentacją serwera Apache:

```
RewriteBase /xyz
RewriteRule ^oldstuff\.html$ newstuff.html
```

W wyniku wykonania powyższych zapisów, żądanie skierowane do adresu */xyz/oldstuff.html* zostaje skierowane do fizycznego pliku */abc/def/newstuff.html*.[3] Cały proces przebiega zaś następująco:

1. Serwer odbiera odwołanie do adresu */xyz/oldstuff.html*.
2. Adres zostaje wewnętrznie poddany następującym przekształceniom:

```
/xyz/oldstuff.html      ->    /abc/def/oldstuff.html (globalna
                              dyrektywa Alias)
/abc/def/oldstuff.html  ->    /abc/def/newstuff.html (regu³a
                              translacji w pliku .htaccess)
/abc/def/newstuff.html  ->    /xyz/newstuff.html (dyrektywa
                              RewriteBase w pliku .htaccess)
/xyz/newstuff.html      ->    /abc/def/newstuff.html (globalna
                              dyrektywa Alias)
```

3. Wynikiem translacji jest adres */abc/def/newstuff.html*.

[3] Aby uniknąć nieporozumień, warto jeszcze zacytować inny fragment dokumentacji: „[…] fizycznym katalogiem odpowiadającym ścieżce */xyz* jest */abc/def*; innymi słowy, plik konfiguracji serwera zawiera dyrektywę `Alias /xyz /abc/def`" – przyp. tłum.

RewriteCond

```
RewriteCond ³añcuch wzorzec
U¿ycie: konfiguracja g³ówna, serwery wirtualne, katalogi
```

Reguła translacji zdefiniowana dyrektywą `RewriteRule` może być poprzedzona jedną lub kilkoma dyrektywami `RewriteCond`, określającymi warunki, które należy spełnić aby translacja została wykonana. Parametr *łańcuch* zawiera specyfikacje zmiennych serwera zapisane w postaci `%{NAZWA_ZMIENNEJ}`. Bieżąca wartość zadanej w ten sposób zmiennej porównywana jest z wyrażeniem regularnym zapisanym w parametrze *wzorzec*; jeśli porównanie wypadnie niepomyślnie, translacja zadana dyrektywą `RewriteRule` nie jest wykonywana.

`NAZWA_ZMIENNEJ` może przyjmować jedną z poniższych wartości:

`API_VERSION`	`PATH_INFO`	`SERVER_PROTOCOL`
`AUTH_TYPE`	`QUERY_STRING`	`SERVER_SOFTWARE`
`DOCUMENT_ROOT`	`REMORE_ADDR`	`THE_REQUEST`
`ENV:` *`dowolna-zmienna`*	`REMOTE_HOST`	`TIME`
`HTTP_ACCEPT`	`REMOTE_USER`	`TIME_DAY`
`HTTP_COOKIE`	`REMOTE_IDENT`	`TIME_HOUR`
`HTTP_FORWARDED`	`REQUEST_FILENAME`	`TIME_MIN`
`HTTP_HOST`	`REQUEST_METHOD`	`TIME_MON`
`HTTP_PROXY_CONNECTION`	`REQUEST_URI`	`TIME_SEC`
`HTTP_REFERER`	`SCRIPT_FILENAME`	`TIME_WDAY`
`HTTP_USER_AGENT`	`SERVER_ADMIN`	`TIME_YEAR`
`HTTP:` *`dowolne-pole`*	`SERVER_NAME`	
`IS_SUBREQ`	`SERVER_PORT`	

Zmienne te odpowiadają określonym polom nagłówka HTTP, zmiennym w kodzie źródłowym programu Apache i wartości bieżącego czasu.

RewriteRule

```
RewriteRule wzorzec podstawienie [opcje]
U¿ycie: konfiguracja g³ówna, serwery wirtualne, katalogi
```

Dyrektywa `RewriteRule` definiuje pojedynczą regułę translacji i może być użyta dowolną liczbę razy. Translacja wykonywana jest „potokowo”, tj. kolejne reguły stosowane są do przetworzenia wyników użycia poprzednich reguł (oznacza to, że kolejność zapisu jest istotna). Użycie danej warunkowane jest porównaniem adresu URL z wyrażeniem regularnym *wzorzec*; jeśli porównanie wypadnie pozytywnie, dokonywane jest *podstawienie*. Działanie reguły może być zmodyfikowane za pomocą argumentu *opcje*, którego dopuszczalne wartości podajemy poniżej (opcje mogą być zapisywane również w formie skróconej).

`redirect|R`
: Wymusza przeadresowanie.

`proxy|P`
Kieruje żądanie do serwera pośredniczącego.

`last|L`
Wymusza zakończenie przetwarzania (reguła traktowana jest jako ostatnia w grupie).

`chain|C`
Jeśli warunek danej reguły jest spełniony, powoduje wykonanie następnej reguły (lub reguł); w przeciwnym przypadku są one pomijane.

`type|T=`*`typ-mime`*
Wymusza dany *typ MIME* dla pliku docelowego.

`nosubreq|NS`
Wymusza pominięcie danej reguły dla podżądań generowanych wewnętrznie przez serwer.

`env|E=`*`zmienna`*`:`*`wartość`*
Definiuje *zmienną* środowiskową, nadając jej określoną *wartość*.

`qsappend|QSA`
Nakazuje dołączenie łańcucha zapytania do przetwarzanego adresu.

`passthrough|PT`
Wymusza podstawienie nazwy pliku do pola identyfikatora URI w wewnętrznej strukturze opisującej żądanie[4] (umożliwia to przetwarzanie przetłumaczonego adresu przez inne dyrektywy, np. `Alias`).

`skip|S=`*`liczba`*
Wymusza pominięcie zadanej *liczby* reguł.

`next|N`
Wymusza ponowne rozpoczęcie przetwarzania od pierwszej reguły.

`gone|G`
Powoduje przesłanie odpowiedzi HTTP o kodzie 410 („URL Gone” – adres został zlikwidowany).

`forbidden|F`
Powoduje przesłanie odpowiedzi HTTP o kodzie 403 („URL Forbidden” – brak zezwolenia na dostęp).

Dla przykładu załóżmy, że chcemy poddawać translacji adresy URL postaci

```
/Jêzyk/~Nazwisko/.../Plik
```

do formy

```
/u/NazwaU¿ytkownika/...Plik.Jêzyk
```

Zdefiniowaną w ten sposób mapę podstawień należy zapisać w pliku */gdziestam/mapa.podstawien*, a następnie dopisać do pliku konfiguracji serwera następujące wiersze:

```
RewriteLog /katalog/rewrite.log
RewriteMap real-to-user txt:/gdziestam/mapa.podstawien
RewriteRule ^/([^/]+)/~([^/]+)/(.*)$
/u/${real-to-user:$2|nobody}/$3.$1
```

[4] Opisująca żądanie struktura `request_rec` jest omówiona w rozdziale 14 – przyp. tłum.

Przykład zastosowania translacji adresów

Pracownicy działu handlowego firmy Butterthlies Polska solidnie przykładają się do pracy, a wachlarz oferowanych produktów rozrósł się do tego stopnia, że stary katalog w postaci pojedynczej strony w języku HTML jest już dla nich za ciasny. Informatycy firmy stworzyli zatem bazę danych zawierającą informacje o pocztówkach, które można pobierać za pomocą prostego programu *cardinfo*, wywoływanego w sposób następujący:

```
cardinfo numer parametr
```

gdzie *numer* to numer wzoru pocztówki, zaś *parametr* może przyjmować wartość „cena", „autor" lub „rozmiar". Niestety nasi handlowcy są zbyt zajęci innymi sprawami, by zawracać sobie głowę zapamiętywaniem składni, toteż postanowiono ułatwić im życie i umożliwić odwoływanie się do bazy za pomocą odpowiednio sformułowanych adresów URL. I tak odwołanie do adresu *http://sales.butterthlies.com.pl/info/2949/cena* powinno zwrócić cenę kartki o numerze wzoru 2949. Odpowiedni plik konfiguracji serwera znajdziesz w katalogu *site.rewrite*:

```
User webuser
Group webgroup
# Apache wymaga podania nazwy serwera, chocia¿ tutaj nie jest
# u¿ywana. Podana nazwa jest u¿ywana domyœlnie dla serwerów nie
# odpowiadaj'cych sekcji VirtualHost.
ServerName www.butterthlies.com.pl

NameVirtualHost 192.168.123.2

<VirtualHost www.butterthlies.com.pl>
ServerAdmin sales@butterthlies.com.pl
DocumentRoot /usr/www/site.rewrite/htdocs/customers
ServerName www.butterthlies.com.pl
ErrorLog /usr/www/site.rewrite/logs/customers/error_log
TransferLog /usr/www/site.rewrite/logs/customers/access_log
</VirtualHost>

<VirtualHost sales.butterthlies.com.pl>
ServerAdmin sales_mgr@butterthlies.com.pl
DocumentRoot /usr/www/site.rewrite/htdocs/salesmen
Options ExecCGI indexes
ServerName sales.butterthlies.com.pl
ErrorLog /usr/www/site.rewrite/logs/salesmen/error_log
TransferLog /usr/www/site.rewrite/logs/salesmen/access_log
RewriteEngine on
RewriteLog logs/rewrite
RewriteLogLevel 9
RewriteRule ^/info/([^/]+)/([^/]+)$ /cgi-bin/cardinfo?$2+$1 [PT]
ScriptAlias /cgi-bin /usr/www/cgi-bin
</VirtualHost>
```

Prawdziwy program *cardinfo* byłby zapewne całkiem skomplikowanym narzędziem, jednak w naszym ćwiczeniu posłużymy się banalnie prostą atrapą:

```
#!/bin/sh
#
echo "content-type:text/html"
echo
echo "Zapyta³eœ o parametr $1 dla pocztówki $2"
```

Aby nie pozostawiać niczego przypadkowi, przed wypróbowaniem naszego systemu wyłączmy translację dyrektywą `RewriteEngine off`. Próba odwołania się do adresu *http://sales.butterthlies.com.pl/info/2949/cena* powinna w takiej sytuacji zakończyć się komunikatem

```
The requested URL /info/2949/cena was not found on this server.
```

I nic dziwnego. Włączmy teraz translację adresów dyrektywą `RewriteEngine on` i przyjrzyjmy się kluczowemu elementowi całego mechanizmu, którym jest następujący wiersz w pliku konfiguracji serwera:

```
RewriteRule ^/info/([^/]+)/([^/]+)$ /cgi-bin/cardinfo?$2+$1 [PT]
```

W ludzkim języku znaczy to mniej więcej tyle: szukając od początku łańcucha, znajdź ciąg znaków `/info/`, po którym następuje jeden lub więcej znaków różnych od „/" i wstaw te znaki do zmiennej `$1` (operację tę realizują nawiasy; dla pierwszego ich zestawu używana jest zmienna `$1`). Następnie wyszukaj znak „/", a po nim kolejny ciąg znaków różnych od „/" i wstaw je do zmiennej `$2`. Wyszukaj koniec łańcucha i przekaż wynik całej operacji (co nakazuje opcja `[PT]`) do następnej dyrektywy, którą jest `ScriptAlias`. Finalny efekt całej tej operacji powinien dać odwołanie do adresu *http://sales.butterthlies.com.pl/cgi-bin/cardinfo?<numer>+<parametr>*.

Gdyby skrypt z jakichś powodów znajdował się na innym serwerze WWW, nasza reguła miałaby postać:

```
RewriteRule ^/info/([^/]+)/([^/]+)$ http://gdzies.tam.com/cgi-bin/
cardinfo?$2+$1 [PT]
```

Zauważmy przy tym, że powyższy wzorzec nie będzie odpowiadał łańcuchowi */info/123/cena/pawel*, gdyż ten ostatni zawiera zbyt wiele znaków „/".

Po uruchomieniu serwera poleceniem `./go` i odwołaniu się za pomocą przeglądarki do adresu *http://sales.butterthlies.com.pl/info/2949/cena* powinniśmy uzyskać odpowiedź:

```
Zapyta³eœ o parametr cena dla pocztówki 2949
```

Korygowanie adresów

Pakiet Apache zawiera jeszcze jeden użyteczny moduł o nazwie *mod_speling*[5]. Umożliwia on automatyczne poprawianie pisowni wielkich i małych liter, uzupełnianie opuszczonych znaków, zamianę znaków błędnie wpisanych i poprawianie przestawień (tzw. „czeskich błędów") w częściach adresów URL odnoszących się do plików i katalogów (odbywa się to poprzez odnoszenie nazw do zawartości systemu plików). Moduł *mod_speling* nie koryguje źle podanych nazw użytkowników.

CheckSpelling

```
CheckSpelling on|off
Wartoœæ domyœlna: off
U¿ycie: wszêdzie
```

[5] Tak, to nie pomyłka. Podejrzewamy, że to jeszcze jeden przejaw humoru programistów (*spelling* (przez dwa „l") = poprawianie błędów pisowni – przyp. tłum.).