



Wdrażanie (deployment) projektów Symfony2 przy pomocy Capifony

Wydanie 1.0

Jacek Siciarek

April 06 2014

1	Spis treści	2
1.1	Narzędzie do synchronizacji katalogów <code>rsync</code>	2
1.2	Capistrano - Swiss Army Knife	5
1.3	Inicjalizacja środowiska produkcyjnego aplikacji	12
1.4	Sprawdzanie ustawień na serwerze lokalnym i zdalnym	14
1.5	Source Control Management	16
1.6	Zakres informacji zwracany przez <code>cap</code>	18
1.7	Capifony - Capistrano dla Symfony	23
1.8	Ustawienie odpowiednich praw dostępu do katalogów	24
1.9	Elementy specyficzne dla Symfony 2	25
1.10	Zdefiniowane zadania <code>tasks</code> i wyzwalacze <code>triggers</code>	30
1.11	Włączanie i wyłączanie aplikacji	33
1.12	Przykładowy, produkcyjny plik <code>Capfile</code>	37
1.13	Narzędzia dodatkowe <code>ant</code>	42



Dokumentacja dostępna pod adresem <https://github.com/siciarek/capifony-4developers>

Spis treści

1.1 Narzędzie do synchronizacji katalogów `rsync`

Rsync jest aplikacją do synchronizacji katalogów na serwerze lokalnym i zdalnym, w obu kierunkach.

Strona projektu <http://rsync.samba.org>.

Szczegółowa dokumentacja <http://rsync.samba.org/ftp/rsync/rsync.html>.

1.1.1 Przykład sparametryzowanej komendy `rsync`

```
SOURCE=.  
TARGET=dude@target.server.com:/var/www/target.server.com
```

```
rsync \  
--itemize-changes \  
--verbose \  
--human-readable \
```

```
--omit-dir-times \
--times \
--perms \
--progress \
--stats \
--compress \
--recursive \
--links \
--delete \
--exclude-from="config/rsync_exclude.txt" \
--dry-run \
$SOURCE $TARGET
```

1.1.2 Przykładowy wynik działania komendy `rsync`

```
sending incremental file list
.L..t..... lib/doctrine/linked_schema.yml -> ../../data/linked_schema.yml
*deleting    web/apc.php
<f.st..... web/import.php

Number of files: 10047
Number of files transferred: 1
Total file size: 150.32M bytes
Total transferred file size: 2.76K bytes
Literal data: 0 bytes
Matched data: 0 bytes
File list size: 210.08K
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 211.97K
Total bytes received: 1.49K
```

```
sent 211.97K bytes  received 1.49K bytes  9.08K bytes/sec
total size is 150.32M  speedup is 704.24 (DRY RUN)
```

1.1.3 Jak rozumieć format opisu zmiany

```
*deleting    web/apc.php
<f+++++++++ newfile.txt
.L..t..... lib/doctrine/linked_schema.yml -> ../../data/linked_schema.yml
<f.st..... web/import.php
```

Format opisu

YXcstpoguax

Gdzie

Y

- < - plik został wysłany na **serwer zdalny**.
- > - plik został pobrany ze **zdalnego serwera**.
- c - element został lokalnie utworzony np. nowy katalog, lub link symboliczny.
- h - oznacza, że dany element jest twardym linkiem (wymaga argumentu `--hard-links`).
- . - plik nie został zmieniony, ale zmieniły się inne parametry pliku.
- * - prefiks dalszej informacji (np. “deleting”).

X

- f** - plik
- d** - katalog
- L** - link symboliczny
- D** - urządzenie

S - plik specjalny (np. nazwane gniazdo (named socket) lub kolejka fifo).

Pozostałe znaki są literami opisującymi zmienione atrybuty pliku lub ”.”, jeżeli dany atrybut nie został zmieniony, poza trzema wyjątkami:

- nowoutworzony plik zamienia każdą z liter na znak “+”
- identyczny element zmienia kropki na spacje
- nieznaną atrybut zmieniany jest na znak ”?” (to się może zdarzyć jeżeli na zdalnym serwerze jest starsza wersja `rsync`)

cstpoguax

A **c** means either that a regular file has a different checksum (requires `–checksum`) or that a symlink, device, or special file has a changed value. Note that if you are sending files to an `rsync` prior to 3.0.1, this change flag will be present only for checksum-differing regular files.

A **s** means the size of a regular file is different and will be updated by the file transfer.

A **t** means the modification time is different and is being updated to the sender’s value (requires `–times`). An alternate value of **T** means that the modification time will be set to the transfer time, which happens when a file/symlink/device is updated without `–times` and when a symlink is changed and the receiver can’t set its time. (Note: when using an `rsync` 3.0.0 client, you might see the **s** flag combined with **t** instead of the proper **T** flag for this time-setting failure.)

A **p** means the permissions are different and are being updated to the sender’s value (requires `–perms`).

An **o** means the owner is different and is being updated to the sender’s value (requires `–owner` and super-user privileges).

A **g** means the group is different and is being updated to the sender’s value (requires `–group` and the authority to set the group).

The **u** slot is reserved for future use.

The **a** means that the ACL information changed.

The **x** means that the extended attribute information changed.

Można użyć argumentu `--out-format` aby dostosować wyjście do swoich potrzeb.

1.2 Capistrano - Swiss Army Knife

`Capistrano` jest napisanym w języku `ruby` systemem zdalnego zarządzania zasobami serwerów, przeznaczonym dla dowolnego języka programowania. Może służyć do wdrażania lub zdalnego wykonywania komend na wielu serwerach jednocześnie.

System powstał z myślą o wdrażaniu projektów `ruby on rails`, jednak jego funkcjonalność jest na tyle uniwersalna, że może służyć właściwie dla każdego języka programowania i każdego typu projektu.

1.2.1 Instalacja

Capistrano jest dostępny jako pakiet `gem` dla `ruby`.

```
$ gem -v
1.8.23
```

Najniższa wymagana wersja to `1.3.x`, jeżeli nie posiadasz takiej, należy przeprowadzić aktualizację a następnie wykonać komendę:

```
$ gem install capistrano
```

Po pomyślnie wykonanej instalacji powinno być możliwe wykonanie poniższej komendy:

```
$ cap --help
Usage: cap [options] action ...
  -d, --debug                Prompts before each remote command execution.
  -e, --explain TASK         Displays help (if available) for the task.
  -F, --default-config       Always use default config, even with -f.
  -f, --file FILE            A recipe file to load. May be given more than once.
  -H, --long-help            Explain these options and environment variables.
  -h, --help                 Display this help message.
  -l [STDERR|STDOUT|file],  Choose logger method. STDERR used by default.
    --logger
  -n, --dry-run              Prints out commands without running them.
  -p, --password              Immediately prompt for the password.
  -q, --quiet                Make the output as quiet as possible.
  -r, --preserve-roles       Preserve task roles
  -S, --set-before NAME=VALUE Set a variable before the recipes are loaded.
  -s, --set NAME=VALUE       Set a variable after the recipes are loaded.
  -T, --tasks [PATTERN]     List all tasks (matching optional PATTERN) in the loaded recipe files.
```


-t, --tool	Abbreviates the output of -T for tool integration.
-V, --version	Display the Capistrano version, and exit.
-v, --verbose	Be more verbose. May be given more than once.
-X, --skip-system-config	Don't load the system config file (capistrano.conf)
-x, --skip-user-config	Don't load the user config file (.caprc)

1.2.2 Skrypt cap

Skrypt cap jest narzędziem umożliwiającym połączenie ze zdalnymi serwerami i wykonywanie na nich komend.

```
$ cat `which cap`  
#!/usr/bin/ruby1.9.1  
#  
# This file was generated by RubyGems.  
#  
# The application 'capistrano' is installed as part of a gem, and  
# this file is here to facilitate running it.  
#  
  
require 'rubygems'  
  
version = ">= 0"  
  
if ARGV.first  
  str = ARGV.first  
  str = str.dup.force_encoding("BINARY") if str.respond_to? :force_encoding  
  if str =~ /\A_(.*)_\z/  
    version = $1  
    ARGV.shift  
  end  
end  
end
```

```
gem 'capistrano', version  
load Gem.bin_path('capistrano', 'cap', version)
```

Lista dostępnych komend po wydaniu polecenia

```
$ cap -T  
$ cap -vT  
$ cap -T doctrine
```

Skryptu można używać z linii komend podając odpowiednie parametry:

```
$ cap invoke HOSTS="new.ses-control.com,platform.ses-support.com" COMMAND="date"  
  
* 2014-04-03 10:36:44 executing `invoke`  
* executing multiple commands in parallel  
  -> "else" :: "date"  
  -> "else" :: "date"  
  servers: ["new.ses-control.com", "platform.ses-support.com"]  
  [new.ses-control.com] executing command  
  [platform.ses-support.com] executing command  
** [out :: new.ses-control.com] czw, 3 kwi 2014, 10:36:44 CEST  
** [out :: platform.ses-support.com] czw, 3 kwi 2014, 10:36:44 CEST  
command finished in 44ms
```

Jeżeli do komendy wymagane jest sudo przekazujemy dodatkowy parametr:

```
$ cap invoke HOSTS="new.ses-control.com,platform.ses-support.com" COMMAND="date" SUDO=1
```

Tu niestety może pojawić się błąd:

```
* 2014-04-03 10:51:12 executing `invoke`  
* executing multiple commands in parallel  
  -> "else" :: "sudo -p 'sudo password: ' date"  
  -> "else" :: "sudo -p 'sudo password: ' date"
```

```
servers: ["new.ses-control.com", "platform.ses-support.com"]
[platform.ses-support.com] executing command
[new.ses-control.com] executing command
*** [err :: platform.ses-support.com] sudo
*** [err :: platform.ses-support.com] :
*** [err :: platform.ses-support.com] no tty present and no askpass program specified
*** [err :: platform.ses-support.com]
*** [err :: new.ses-control.com] sudo
*** [err :: new.ses-control.com] :
*** [err :: new.ses-control.com] no tty present and no askpass program specified
*** [err :: new.ses-control.com]
    command finished in 87ms
failed: "sh -c 'sudo -p '\\\\'sudo password: '\\\\' date'" on new.ses-control.com,platform.ses-support.com
```

Aby umożliwić wykonywanie komend sudo musimy utworzyć plik konfiguracyjny i dodać linię:

```
default_run_options[:pty] = true
```

Poniżej domyślne nazwy plików konfiguracyjnych dla cap.

/etc/capistrano.conf - ustawienie globalne serwera (pomijany parametrem --skip-system-config)

~/.caprc file - ustawienie dla konta użytkownika (pomijany parametrem --skip-user-config)

./Capfile - ustawienie dla projektu

deploy.rb - alternatywne ustawienie dla projektu

1.2.3 Automatyczne tworzenie pliku konfiguracji

Plik konfiguracyjny dla projektu można utworzyć automatycznie:

```
$ cd /path/to/your/project
$ capify .
```

```
[add] writing '/path/to/your/project/Capfile'
[add] making directory '/path/to/your/project/config'
[add] writing '/path/to/your/project/config/deploy.rb'
[done] capified!
```

Zawartość katalogu /path/to/your/project:

```
$ ls -R /path/to/your/project
/path/to/your/project:
Capfile  config
```

```
/path/to/your/project/config:
deploy.rb
```

Zawartość /path/to/your/project/Capfile

```
load 'deploy'
# Uncomment if you are using Rails' asset pipeline
# load 'deploy/assets'
load 'config/deploy' # remove this line to skip loading any of the default tasks
```

Zawartość /path/to/your/project/deploy.rb

```
set :application, "set your application name here"
set :repository,  "set your repository location here"
```

```
# set :scm, :git # You can set :scm explicitly or Capistrano will make an intelligent guess based on known version control systems
# Or: 'accurev', 'bazaar', 'cvs', 'darcs', 'git', 'mercurial', 'perforce', 'subversion' or 'none'
```

```
role :web, "your web-server here" # Your HTTP server, Apache/etc
role :app, "your app-server here" # This may be the same as your 'Web' server
```

```
role :db, "your primary db-server here", :primary => true # This is where Rails migrations will run
role :db, "your slave db-server here"

# if you want to clean up old releases on each deploy uncomment this:
# after "deploy:restart", "deploy:cleanup"

# if you're still using the script/reaper helper you will need
# these http://github.com/rails/irs_process_scripts

# If you are using Passenger mod_rails uncomment this:
# namespace :deploy do
#   task :start do ; end
#   task :stop do ; end
#   task :restart, :roles => :app, :except => { :no_release => true } do
#     run "#{try_sudo} touch #{File.join(current_path, 'tmp', 'restart.txt')}"
#   end
# end
```

1.2.4 Przykładowa konfiguracja w pliku Capfile

```
load 'deploy' if respond_to?(:namespace) # cap2 differentiator

# RUN OPTIONS:

default_run_options[:pty] = true

# SSH SETTINGS:

ssh_options[:forward_agent] = true

# SERVERS
```

```
role :app,  
  'new.ses-control.com',  
  'platform.ses-support.com'
```

Powyższa konfiguracja umożliwia wykonanie komend bez podawania listy serwerów.

```
$ cap invoke COMMAND="date"  
$ cap invoke COMMAND="date" SUDO=1
```

1.3 Inicjalizacja środowiska produkcyjnego aplikacji

Przykładowa konfiguracja w Capfile

```
set :application, "myapp"  
set :user, "dude"  
set :domain, "target.server.com"  
set :repository, "git@myscm.server.net:/home/git/repos/#{application}.git"  
set :deploy_to, "/var/www/#{domain}"  
  
role :app, :domain, :primary => true # This may be the same as your ``Web`` server  
role :web, :domain # Your HTTP server, Apache/etc
```

1.3.1 Przygotowanie środowiska produkcyjnego

```
$ cap deploy:setup
```

```
* 2014-04-04 12:54:57 executing `deploy:setup`  
* executing multiple commands in parallel  
  -> "else" :: "sudo -p 'sudo password: ' mkdir -p /var/www/target.server.com /var/www/target.server.com/releases /var
```

```
-> "else" :: "sudo -p 'sudo password: ' mkdir -p /var/www/target.server.com /var/www/target.server.com/releases /var
-> "else" :: "sudo -p 'sudo password: ' mkdir -p /var/www/target.server.com /var/www/target.server.com/releases /var
-> "else" :: "sudo -p 'sudo password: ' mkdir -p /var/www/target.server.com /var/www/target.server.com/releases /var
servers: ["new.ses-control.com", "platform.ses-support.com", "integration.sescom.pl", "hm.ses-control.com"]
[integration.sescom.pl] executing command
[hm.ses-control.com] executing command
[platform.ses-support.com] executing command
[new.ses-control.com] executing command
command finished in 130ms
* executing multiple commands in parallel
-> "else" :: "sudo -p 'sudo password: ' chmod g+w /var/www/target.server.com /var/www/target.server.com/releases /va
-> "else" :: "sudo -p 'sudo password: ' chmod g+w /var/www/target.server.com /var/www/target.server.com/releases /va
-> "else" :: "sudo -p 'sudo password: ' chmod g+w /var/www/target.server.com /var/www/target.server.com/releases /va
-> "else" :: "sudo -p 'sudo password: ' chmod g+w /var/www/target.server.com /var/www/target.server.com/releases /va
servers: ["new.ses-control.com", "platform.ses-support.com", "integration.sescom.pl", "hm.ses-control.com"]
[integration.sescom.pl] executing command
[hm.ses-control.com] executing command
[platform.ses-support.com] executing command
[new.ses-control.com] executing command
command finished in 98ms
```

Jeżeli wszystkie prawa dostępu będą ustawione prawidłowo na **serwerze produkcyjnym** zostanie utworzona poniższa struktura katalogów:

```
/var/www/target.server.com
|-- releases
`-- shared
```

1.3.2 Sprawdzenie konfiguracji serwera produkcyjnego

```
cap deploy:check
```

1.3.3 Przykład zawartości katalogu aplikacji po kolejnym wdrożeniu

```
`-- /var/www/target.server.com
|-- current -> /var/www/target.server.com/releases/20100512131539
|-- releases
|   |-- 20100512131539
|   |-- 20100509150741
|   |-- 20100509145325
`-- shared
    |-- web
    |   |-- uploads
    |-- logs
    |-- vendors
    |-- config
    |-- app
    |-- parameters.yml
```

1.4 Sprawdzanie ustawień na serwerze lokalnym i zdalnym

Aby umożliwić sprawdzenie ustawień na lokalnym i zdalnym serwerze Capistrano udostępnia polecenie

```
$ cap deploy:check
```

W wersji podstawowej sprawdza obecność i prawa dostępu do katalogów utworzonych poleceniem

```
$ cap deploy:setup
```

Dostępność komendy `tar` na zdalnym serwerze oraz programu do kontroli wersji na lokalnym.

Wyświetlane są tylko efekty działania komendy sprawdzającej na zdalnym serwerze, chyba, że pojawią się błędy, wtedy wyświetlane są w postaci komunikatu np.:

The following dependencies failed. Please check them and try again:

```
--> 'git' could not be found in the path on the local host  
--> 'setfacl' could not be found in the path (ekk.sescom.pl)
```

Jeżeli sprawdzanie zakończy się sukcesem pojawi się komunikat:

```
You appear to have all necessary dependencies installed
```

Aby umożliwić dodawanie sprawdzania zasobów Capistrano udostępnia metodę `depend` Przyjmującą 3 argumenty

Pierwszy parametr określa miejsce sprawdzania, może przyjmować poniższe wartości:

- `:local` jeżeli chcemy sprawdzać serwer z którego przeprowadzamy wdrożenie
- `:remote` jeżeli chcemy sprawdzić zdalne serwery.

Drugi parametr określa typ sprawdzanego zasobu i może przyjmować poniższe wartości:

- `:directory` sprawdza czy istnieje katalog.
- `:file` sprawdza czy istnieje plik.
- `:writable` sprawdza czy katalog lub plik posiada prawa do zapisu.
- `:command` sprawdza czy komenda jest dostępna.
- `:deb` sprawdza czy pakiet jest zainstalowany.

Trzeci parametr to nazwa sprawdzanego zasobu w postaci łańcucha znaków.

1.4.1 Przykłady użycia

```
depend :local, :command, "convert"  
depend :remote, :command, "setfacl"
```

1.5 Source Control Management

Przykładowa konfiguracja w Capfile

```
set :application, "myapp"
set :user, "dude"
set :domain, "target.server.com"
set :repository, "git@myscm.server.net:/home/git/repos/#{application}.git"
set :deploy_to, "/var/www/#{domain}"

role :app, :domain, :primary => true # This may be the same as your ``Web`` server
role :web, :domain # Your HTTP server, Apache/etc

set :scm, :git
set :deploy_via, :copy
```

1.5.1 Wartości :scm

:none System kontroli wersji nie jest stosowany

:scm System kontroli wersji jest wykrywany automatycznie

:git GIT <http://git-scm.com>

:accurev AccuRev <http://www.accurev.com>

:bazaar Bazaar <http://bazaar.canonical.com>

:cvs CVS <http://www.nongnu.org/cvs/>

:darcs Darcs <http://darcs.net>

:subversion SVN <http://subversion.tigris.org>, <http://subversion.apache.org>

:mercurial Mercurial <http://mercurial.selenic.com>

:perforce PERFORCE <http://www.perforce.com>

1.5.2 Wartości :deploy_via

:copy Na **serwerze źródłowym** wykonuje eksport wersji, domyślnie, do katalogu `/tmp`, następnie kopiuje jego zawartość na **serwer produkcyjny**.

```
set :copy_dir, "./tmp" # optional

set :copy_exclude, [
  ".git",
  ".gitignore",
  "/app/config/config_test.yml",
  "/bin",
  "/src/Application/MainBundle/Tests",
  "/src/Application/MainBundle/Resources/doc",
  "/web/apple-touch-icon.png",
]
```

:checkout Wykonuje operację checkout na **serwerze produkcyjnym**, nie zalecany, ze względu na możliwość nieautoryzowanej modyfikacji zawartości repozytorium.

:export Wykonuje operację export na **serwerze produkcyjnym**.

:remote_cache Wykonuje `git pull` (lub odpowiednik tej komendy w innych SCM) w katalogu `shared/cached_copy`, zamiast zaciągania całego repozytorium. Ma znaczenie jeżeli zależy nam na przyspieszeniu procesu wdrożenia. Jeżeli w czasie wdrożenia pojawią się problemy należy usunąć katalog `shared/cached_copy`.

:rsync_with_remote_cache W tym przypadku `rsync` utworzy katalog na **serwerze produkcyjnym** i będzie go synchronizował ze zmianami w repozytorium. Wymaga instalacji dodatkowego `gema`.

1.5.3 Wartości specyficzne dla danego SCM

Możemy dodać do konfiguracji dodatkowe parametry, specyficzne dla użytego SCM np.

```
set :git_enable_submodules, true
```

1.6 Zakres informacji zwracany przez cap

Capistrano umożliwia wyświetlanie informacji o wykonywanych komendach w różnych zakresach.

1.6.1 Typy zakresu

```
logger.level = Logger::IMPORTANT # -->
logger.level = Logger::INFO      # **
logger.level = Logger::DEBUG     # *
logger.level = Logger::TRACE     #
logger.level = Logger::MAX_LEVEL # Logger::TRACE = Logger::MAX_LEVEL
```

Fragment kodu logger.rb

```
module Capistrano
  class Logger # :nodoc:
    attr_accessor :level
    attr_reader   :device

    IMPORTANT = 0
    INFO      = 1
    DEBUG     = 2
    TRACE     = 3
  end
end
```

```
MAX_LEVEL = 3
```

1.6.2 Przykładowy wynik ustawień

```
logger.level = Logger::IMPORTANT
```

```
--> Updating code base with copy strategy
--> Creating cache directory.....V
--> Creating symlinks for shared directories.....V
--> Creating symlinks for shared files.....V
--> Normalizing asset timestamps.....V
--> Downloading Composer.....V
```

```
logger.level = Logger::INFO`
```

```
** sftp upload parameters.yml -> /d0/www/platform-integration/shared/app/config/parameters.yml
** transaction: start
--> Updating code base with copy strategy
** sftp upload /tmp/20140319145926.tar.gz -> /tmp/20140319145926.tar.gz
--> Creating cache directory
--> Creating symlinks for shared directories
--> Creating symlinks for shared files
--> Normalizing asset timestamps
--> Downloading Composer
** [out :: myspec.pl] #!/usr/bin/env php
** [out :: myspec.pl] Some settings on your machine may cause stability issues with Composer.
** [out :: myspec.pl] If you encounter issues, try to change the following:
** [out :: myspec.pl]
** [out :: myspec.pl] Your PHP (5.3.3-7+squeezel4) is quite old, upgrading to PHP 5.3.4 or higher is recommended.
** [out :: myspec.pl] Composer works with 5.3.2+ for most people, but there might be edge case issues.
** [out :: myspec.pl]
```

```
** [out :: myspec.pl] Downloading...
** [out :: myspec.pl]
** [out :: myspec.pl] Composer successfully installed to: /d0/www/platform-integration/releases/20140319145926/compos
** [out :: myspec.pl]
** [out :: myspec.pl] Use it: php composer.phar
--> Updating Composer dependencies
** [out :: myspec.pl] Loading composer repositories with package information
** [out :: myspec.pl] Updating dependencies

logger.level = Logger::DEBUG

* 2014-03-19 16:00:41 executing `deploy`
* 2014-03-19 16:00:41 executing `upload_parameters`
* executing "mkdir -p /d0/www/platform-integration/shared/app/config"
** sftp upload parameters.yml -> /d0/www/platform-integration/shared/app/config/parameters.yml
* sftp upload complete
* 2014-03-19 16:00:41 executing `deploy:update`
** transaction: start
* 2014-03-19 16:00:41 executing `deploy:update_code`
--> Updating code base with copy strategy
* getting (via checkout) revision cc301318ff9b560edd494a57db6c66ea95c878c7 to /tmp/20140319150041
* processing exclusions...
* Compressing /tmp/20140319150041 to /tmp/20140319150041.tar.gz
** sftp upload /tmp/20140319150041.tar.gz -> /tmp/20140319150041.tar.gz
* sftp upload complete
* executing "cd /d0/www/platform-integration/releases && tar xzf /tmp/20140319150041.tar.gz && rm /tmp/20140319150041"
* 2014-03-19 16:00:44 executing `deploy:finalize_update`
* executing "chmod -R g+w /d0/www/platform-integration/releases/20140319150041"
--> Creating cache directory
* executing "sh -c 'if [ -d /d0/www/platform-integration/releases/20140319150041/app/cache ] ; then rm -rf /d0/www/p
* executing "sh -c 'mkdir -p /d0/www/platform-integration/releases/20140319150041/app/cache && chmod -R 0777 /d0/www
* executing "chmod -R g+w /d0/www/platform-integration/releases/20140319150041/app/cache"
```

```
* 2014-03-19 16:00:44 executing `deploy:share_childs`
--> Creating symlinks for shared directories
* executing "mkdir -p /d0/www/platform-integration/shared/app/logs"
* executing "sh -c 'if [ -d /d0/www/platform-integration/releases/20140319150041/app/logs ] ; then rm -rf /d0/www/pl
* executing "ln -nfs /d0/www/platform-integration/shared/app/logs /d0/www/platform-integration/releases/201403191500
* executing "mkdir -p /d0/www/platform-integration/shared/web/uploads"
* executing "sh -c 'if [ -d /d0/www/platform-integration/releases/20140319150041/web/uploads ] ; then rm -rf /d0/www
* executing "ln -nfs /d0/www/platform-integration/shared/web/uploads /d0/www/platform-integration/releases/201403191
* executing "mkdir -p /d0/www/platform-integration/shared/vendor"
* executing "sh -c 'if [ -d /d0/www/platform-integration/releases/20140319150041/vendor ] ; then rm -rf /d0/www/plat
* executing "ln -nfs /d0/www/platform-integration/shared/vendor /d0/www/platform-integration/releases/20140319150041
--> Creating symlinks for shared files
* executing "mkdir -p /d0/www/platform-integration/shared/app/config"
* executing "touch /d0/www/platform-integration/shared/app/config/parameters.yml"
* executing "ln -nfs /d0/www/platform-integration/shared/app/config/parameters.yml /d0/www/platform-integration/rele
--> Normalizing asset timestamps
* executing "find /d0/www/platform-integration/releases/20140319150041/web/css /d0/www/platform-integration/releases
* 2014-03-19 16:00:45 executing `symfony:composer:update`
* 2014-03-19 16:00:45 executing `symfony:composer:get`
* executing "if [ -e /d0/www/platform-integration/releases/20140319150041/composer.phar ]; then echo 'true'; fi"
--> Downloading Composer
* executing "sh -c 'cd /d0/www/platform-integration/releases/20140319150041 && curl -s http://getcomposer.org/install
** [out :: myspec.pl] #!/usr/bin/env php
** [out :: myspec.pl] Some settings on your machine may cause stability issues with Composer.
** [out :: myspec.pl] If you encounter issues, try to change the following:
** [out :: myspec.pl]
** [out :: myspec.pl] Your PHP (5.3.3-7+squeeze14) is quite old, upgrading to PHP 5.3.4 or higher is recommended.
** [out :: myspec.pl] Composer works with 5.3.2+ for most people, but there might be edge case issues.
** [out :: myspec.pl]
** [out :: myspec.pl] Downloading...
** [out :: myspec.pl]
** [out :: myspec.pl] Composer successfully installed to: /d0/www
```

```
logger.level = Logger::TRACE
logger.level = Logger::MAX_LEVEL # Logger::TRACE jest równy Logger::MAX_LEVEL

* 2014-03-19 16:02:46 executing `deploy`
  triggering before callbacks for `deploy`
* 2014-03-19 16:02:46 executing `upload_parameters`
* executing "mkdir -p /d0/www/platform-integration/shared/app/config"
  servers: ["myspec.pl"]
  [myspec.pl] executing command
  command finished in 24ms
  servers: ["myspec.pl"]
** sftp upload parameters.yml -> /d0/www/platform-integration/shared/app/config/parameters.yml
  [myspec.pl] /d0/www/platform-integration/shared/app/config/parameters.yml
  [myspec.pl] done
* sftp upload complete
* 2014-03-19 16:02:46 executing `deploy:update`
** transaction: start
* 2014-03-19 16:02:46 executing `deploy:update_code`
  triggering before callbacks for `deploy:update_code`
--> Updating code base with copy strategy
  executing locally: "git ls-remote git@ses-support.net:/home/git/repos/platform-integration.git HEAD"
  command finished in 91ms
* getting (via checkout) revision cc301318ff9b560edd494a57db6c66ea95c878c7 to /tmp/20140319150246
  executing locally: git clone -q git@ses-support.net:/home/git/repos/platform-integration.git /tmp/20140319150246
  command finished in 2742ms
* processing exclusions...
* Compressing /tmp/20140319150246 to /tmp/20140319150246.tar.gz
  executing locally: tar czf 20140319150246.tar.gz 20140319150246
  command finished in 38ms
  servers: ["myspec.pl"]
** sftp upload /tmp/20140319150246.tar.gz -> /tmp/20140319150246.tar.gz
  [myspec.pl] /tmp/20140319150246.tar.gz
```



```
[myspec.pl] done
* sftp upload complete
* executing "cd /d0/www/platform-integration/releases && tar xzf /tmp/20140319150246.tar.gz && rm /tmp/20140319150246.tar.gz"
  servers: ["myspec.pl"]
  [myspec.pl] executing command
  command finished in 52ms
* 2014-03-19 16:02:49 executing 'deploy:finalize_update'
* executing "chmod -R g+w /d0/www/platform-integration/releases/20140319150246"
  servers: ["myspec.pl"]
  [myspec.pl] executing command
  command finished in 23ms
```

1.6.3 Zapis loga do pliku

```
log_file = "deployment.log"
self.logger = Logger.new(:output => log_file)

logger.level = Logger::DEBUG
```

1.7 Capifony - Capistrano dla Symfony

Capifony jest napisanym w języku ruby systemem zdalnego zarządzania zasobami serwerów, dedykowanego do wdrożeń projektów Symfony.

1.7.1 Instalacja

Capifony jest dostępny jako pakiet gem dla ruby.

```
$ gem -v  
1.8.23
```

Najniższa wymagana wersja to 1.3.x, jeżeli nie posiadasz takiej, należy przeprowadzić aktualizację a następnie wykonać komendę:

```
$ gem install capifony
```

Po pomyślnie wykonanej instalacji powinno być możliwe wykonanie poniższej komendy:

```
$ capifony --help  
Usage: capifony [path]  
  -h, --help                Displays this help info  
  -v, --version              Capify specific symfony version (1|2)  
  -s, --symfony VERSION     Specify app name (folder) to capify  
  -p, --app NAME
```

1.8 Ustawienie odpowiednich praw dostępu do katalogów

```
set :writable_dirs,      [ "app/cache", "app/logs" ]  
set :webserver_user,     "www-data"  
set :permission_method, :acl  
set :use_set_permissions, true
```

Ustawienie ścieżek do których inny użytkownik ma mieć prawo dostępu typu rw.

```
set :writable_dirs,      [ "app/cache", "app/logs", "web/uploads" ]
```

Podanie użytkownika, który ma mieć prawa dostępu typu rw do powyższych katalogów.

```
set :webserver_user,     "www-data"
```

Podanie sposobu w jaki mają być ustawiane prawa dostępu.

```
set :permission_method, :acl
```

Dostępne wartości parametru:

- :acl - korzysta z setfacl
- :chown - wymaga ustawienia set use_sudo:, true
- :chmod - korzysta z chmod +a

Preferowana jest wartość :acl

Flaga, czy należy ustawiać prawa dostępu, może być czasowo wyłączane bez potrzeby usuwania całej powyższej konfiguracji.

```
set :use_set_permissions, true
```

1.9 Elementy specyficzne dla Symfony 2

Nagłówek pliku konfiguracyjnego Capfile dla projektu Symfony 2

```
load 'deploy' if respond_to?(:namespace) # cap2 differentiator

require 'capifony_symfony2'

$ cap -vT
```

1.9.1 Komendy capistrano

```
cap deploy                # Deploys your project.
cap deploy:check           # Test deployment dependencies.
cap deploy:cleanup         # Clean up old releases.
```

```
cap deploy:cold          # Deploys and starts a 'cold' app...
cap deploy:create_symlink # Updates the symlink to the most...
cap deploy:drop          # Drops :deploy_to directory
cap deploy:finalize_update # Updates latest release source path
cap deploy:migrate       # Runs the Symfony2 migrations
cap deploy:migrations    # Deploy and run pending migrations.
cap deploy:pending       # Displays the commits since your...
cap deploy:pending:diff  # Displays the 'diff' since your ...
cap deploy:restart       # Blank task exists as a hook int...
cap deploy:rollback      # Rolls back to a previous versio...
cap deploy:rollback:cleanup # [internal] Removes the most rec...
cap deploy:rollback:code  # Rolls back to the previously de...
cap deploy:rollback:revision # [internal] Points the current s...
cap deploy:set_permissions # Sets permissions for writable_d...
cap deploy:setup         # Prepares one or more servers fo...
cap deploy:share_childs  # Symlinks static directories and...
cap deploy:start         # Blank task exists as a hook int...
cap deploy:stop          # Blank task exists as a hook int...
cap deploy:symlink       # Deprecated API.
cap deploy:test_all      # Deploys the application and run...
cap deploy:update        # Copies your project and updates...
cap deploy:update_code    # Copies your project to the remo...
cap deploy:upload        # Copy files to the currently dep...
cap deploy:web:disable    # Present a maintenance page to v...
cap deploy:web:enable     # Makes the application web-acces...
cap invoke               # Invoke a single command on the ...
cap shared:folder:download # Downloads a backup of the share...
cap shell                # Begin an interactive Capistrano...
```

1.9.2 Komendy dotyczące bazy danych

```
cap database:dump:local      # Dumps local database
cap database:dump:remote    # Dumps remote database
cap database:move:to_local  # Dumps remote database, download...
cap database:move:to_remote # Dumps local database, loads it ...
```

1.9.3 Komendy composera

```
cap symfony:composer:copy_vendors #
cap symfony:composer:dump_autoload # Dumps an optimized autoloader
cap symfony:composer:dump_autoload_temp # Dumps an optimized autoloader
cap symfony:composer:get           # Gets composer and installs it
cap symfony:composer:install       # Runs composer to install vendor...
cap symfony:composer:update        # Runs composer to update vendors...
```

1.9.4 Komendy Symfony 2

```
cap symfony # Runs custom symfony command

cap symfony:assetic:dump # Dumps all assets to the filesystem
cap symfony:assets:install # Installs bundle's assets
cap symfony:assets:update_version # Updates assets version (in conf...

cap symfony:bootstrap:build # Runs the bin/build_bootstrap sc...
cap symfony:cache:clear     # Cache clear
cap symfony:cache:warmup    # Cache warmup

cap symfony:project:clear_controllers # Clears all non production envir...
```

1.9.5 Komendy Doctrine

```
cap symfony:doctrine:cache:clear_metadata # Clears all metadata cache for a...
cap symfony:doctrine:cache:clear_query   # Clears all query cache for a en...
cap symfony:doctrine:cache:clear_result   # Clears result cache for a entit...
cap symfony:doctrine:database:create      # Creates the configured databases
cap symfony:doctrine:database:drop        # Drops the configured databases
cap symfony:doctrine:init:acl              # Mounts ACL tables in the database
cap symfony:doctrine:load_fixtures        # Load data fixtures
cap symfony:doctrine:migrations:migrate   # Executes a migration to a speci...
cap symfony:doctrine:migrations:status    # Views the status of a set of mi...

cap symfony:doctrine:mongodb:indexes:create # Allows you to create indexes *o...
cap symfony:doctrine:mongodb:indexes:drop   # Allows you to drop indexes *onl...
cap symfony:doctrine:mongodb:schema:create  # Allows you to create databases,...
cap symfony:doctrine:mongodb:schema:drop    # Allows you to drop databases, c...
cap symfony:doctrine:mongodb:schema:update  # Allows you to update databases,...
cap symfony:doctrine:schema:create          # Processes the schema and either...
cap symfony:doctrine:schema:drop            # Drops the complete database sch...
cap symfony:doctrine:schema:update          # Updates database schema of Enti...
```

1.9.6 Komendy Propel

```
cap symfony:propel:build:acl              # Generates ACLs models
cap symfony:propel:build:acl_load          # Inserts propel ACL tables
cap symfony:propel:build:all_and_load      # Builds the Model classes, SQL s...
cap symfony:propel:build:model             # Builds the Model classes
cap symfony:propel:build:sql               # Builds SQL statements
cap symfony:propel:build:sql_load          # Inserts SQL statements
cap symfony:propel:database:create         # Creates the configured databases
cap symfony:propel:database:drop          # Drops the configured databases
```

1.9.7 Komendy bin/vendors

```
cap symfony:vendors:install      # Runs the bin/vendors script to ...
cap symfony:vendors:reinstall    # Runs the bin/vendors script to ...
cap symfony:vendors:upgrade      # Runs the bin/vendors script to ...
```

1.9.8 Komendy przeglądania logów

```
cap symfony:logs:tail            # Tail prod.log
cap symfony:logs:tail_dev        # Tail dev.log
```

1.9.9 Uwagi

Przy komendach symfony innych niż zdefiniowane np. `fos:js-routing:dump` pojawia się zapytanie

```
$ cap symfony
* 2014-04-05 21:02:37 executing `symfony'
task_arguments [cache:clear] :
```

aby tego uniknąć, kiedy np. chcemy użyć komendy w skrypcie, musimy przekierować komendę na STDIN.

```
$ echo fos:js-routing:dump | cap symfony
```

Zastosowanie w pliku konfiguracyjnym ant

```
<target name="fosjs">
  <exec executable="bash">
    <arg line="-c &quot;echo fos:js-routing:dump | cap symfony&quot;"/>
  </exec>
</target>
```

1.9.10 Ustawienia konfiguracyjne composera

```
set :use_composer,      true
# set :composer_bin,    "/home/usync/bin/composer"
set :interactive_mode,   false
```

1.9.11 Ustawienia konfiguracyjne specyficzne dla Symfony

```
set :model_manager,     "doctrine" # Or: "propel"
set :shared_files,      [ "app/config/parameters.yml", "build.xml", "properties.cnf", "web/app.php" ]
set :shared_children,    [ log_path, web_path + "/uploads", "vendor" ]

set :update_vendors,     true
set :dump_assetic_assets, true
set :assets_symlinks,    true
```

1.10 Zdefiniowane zadania tasks i wyzwalacze triggers

Zdefiniowane zadania są sposobem na rozszerzenie podstawowych funkcjonalności udostępnianych przez cap.

Najprostszy task prezentujący jedynie strukturę

```
desc 'To jest task testowy.'
task :dummy do
  # do nothing.
end
```

Wersja z bogatszym opisem


```
desc <<-DESC
  To jest task testowy. Task testowy "dummy" \
  służy wyłącznie do prezentacji sposobu budowania tasków \
  i nie powinien zawierać żadnych istotnych elementów.
DESC
task :dummy do
  # do nothing.
end
```

Użycie zdefiniowanego zadania

```
$ cap dummy
```

Bardziej użyteczny task wgrywający `parameters.yml` do katalogu aplikacji.

```
desc <<-DESC
  Kopiuje plik konfiguracyjny aplikacji. Plik konfiguracyjny \
  nie powinien być częścią repozytorium, ponieważ zawiera dane niejawne \
  jak np. hasło dostępowe do serwera bazy danych.
DESC
task :upload_parameters do
  origin_file = "parameters.yml"
  destination_file = shared_path + "/app/config/#{origin_file}" # Notice the shared_path

  try_sudo "mkdir -p #{File.dirname(destination_file)}"
  top.upload(origin_file, destination_file)
end
```

```
before "deploy:share_childs", "upload_parameters"
```

Wgrywanie i usuwanie front controllera debugującego.

```
desc <<-DESC
  Wgraj front kontroler debugujący.
```

```
DESC
task :dbg do
  origin_file = "app_dev.php"
  destination_file = deploy_to + "/current/web/" + origin_file
  top.upload(origin_file, destination_file)
end

desc <<-DESC
  Usun front kontroler debugujący.
DESC
task :undbg do
  origin_file = "app_dev.php"
  destination_file = deploy_to + "/current/web/" + origin_file
  try_sudo "rm -rvf #{destination_file}"
end
```

Działanie powyższego zbliżone jest do `cap symfony:project:clear_controllers`.

1.10.1 Wyzwalacze (triggers)

Wyzwalacze pozwalają na automatyczne uruchamianie zadań wbudowanych lub zdefiniowanych w przypadku zaistnienia odpowiedniego zdarzenia event.

Składnia

```
cut_point, "zadanie_docelowe", "zadanie_do_wykonania_w_kontekscie", "kolejne_zadanie_do_wykonania_w_kontekscie", ...
```

Obecnie capistrano obsługuje poniższe punkty wcięcia `cut_point` (aspect programming domain):

```
# * :before, uruchamia podane zadanie/zadania przed wywołaniem danego zdarzenia docelowego
# * :after, po wywołaniu danego zdarzenia docelowego
# * :start, przed uruchomieniem głównego zadania
# * :finish, po zakończeniu głównego zadania
```

```
# * :load, po załadowaniu wszystkich zadań  
# * :exit, po wykonaniu wszystkich zadań
```

Są jeszcze bardziej wymyślne konstrukcje, do zobaczenia w dokumentacji (w kodzie) capistrano/callback.

1.11 Włączanie i wyłączanie aplikacji

1.11.1 Podstawowe komendy

```
cap deploy:web:disable  
cap deploy:web:enable
```

Zmiana wyświetlanego powodu wyłączenia REASON i czasu ponownego uruchomienia UNTIL

```
cap deploy:web:disable REASON="database migration" UNTIL="2014-04-08"
```

1.11.2 Szablon strony z powiadomieniem

Szablon domyślny /usr/lib/ruby/vendor_ruby/capistrano/recipes/deploy/templates/maintenance.rhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">  
  
<head>  
  <meta http-equiv="content-type" content="text/html; charset=UTF-8" />  
  <title>System down for maintenance</title>
```

```
<style type="text/css">
  div.outer {
    position: absolute;
    left: 50%;
    top: 50%;
    width: 500px;
    height: 300px;
    margin-left: -260px;
    margin-top: -150px;
  }

  .DialogBody {
    margin: 0;
    padding: 10px;
    text-align: left;
    border: 1px solid #ccc;
    border-right: 1px solid #999;
    border-bottom: 1px solid #999;
    background-color: #fff;
  }

  body { background-color: #fff; }
</style>
</head>

<body>

  <div class="outer">
    <div class="DialogBody" style="text-align: center;">
      <div style="text-align: center; width: 200px; margin: 0 auto;">
        <p style="color: red; font-size: 16px; line-height: 20px;">
          The system is down for <%= reason ? reason : "maintenance" %>

```

```

        as of <%= Time.now.strftime("%H:%M %Z") %>.
    </p>
    <p style="color: #666;">
        It'll be back <%= deadline ? deadline : "shortly" %>.
    </p>
</div>
</div>
</div>

</body>
</html>

Szablon zmodyfikowany ./maintenance.pl.rhtml

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">

<head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
    <title>Aplikacja jest czasowo niedostępna</title>

    <style type="text/css">
        div.outer {
            position: absolute;
            left: 50%;
            top: 50%;
            width: 500px;
            height: 300px;
            margin-left: -260px;
            margin-top: -150px;
        }
    </style>

```

```
.DialogBody {
    margin: 0;
    padding: 10px;
    text-align: left;
    border: 1px solid #ccc;
    border-right: 1px solid #999;
    border-bottom: 1px solid #999;
    background-color: #fff;
}

body { background-color: #fff; }
</style>
</head>

<body>

<div class="outer">
    <div class="DialogBody" style="text-align: center;">
        <div style="text-align: center; width: 200px; margin: 0 auto;">
            <p style="color: red; font-size: 16px; line-height: 20px;">
                Od <%= Time.now.strftime("%Y-%m-%d %H:%M") %>
                aplikacja jest czasowo niedostępna
                z powodu <%= reason ? reason : "konserwacji" %>.
            </p>
            <p style="color: #666;">
                Zostanie ponownie uruchomiona <%= deadline ? deadline : "niebawem" %>.
            </p>
        </div>
    </div>
</div>

</body>
```

`</html>`

Aby capistrano korzystał ze zmodyfikowanego szablonu należy dodać do Capfile

```
set :maintenance_template_path, "maintenance.pl.rhtml"
```

```
cap deploy:web:disable REASON="migracji baz danych na nowy serwer" UNTIL="8 kwietnia 2014 o 9:30"
```

1.11.3 Konfiguracja serwerów www

Apache

```
RewriteCond %{DOCUMENT_ROOT}/maintenance.html -f  
RewriteRule .? maintenance.html [L]
```

NginX

```
if (-f /maintenance.html)  
{  
    rewrite .? maintenance.html last;  
}
```

1.12 Przykładowy, produkcyjny plik Capfile

```
load 'deploy' if respond_to?(:namespace) # cap2 differentiator  
  
require 'capifony_symfony2'  
  
#SSH SETTINGS:
```

```
ssh_options[:forward_agent] = true
set :use_sudo,                false

# -----

set :application, Dir.pwd.split('/').last
set :user,         "usync"
set :domain,       "#{application}.sescom.pl"
set :repository,   "git@ses-support.net:/home/git/repos/#{application}.git"
set :deploy_to,    "/d0/www/#{domain}"

# -----

role :web,          "#{domain}"
role :app,           "#{domain}", :primary => true
# SCM SETTINGS:

set :scm,            :git
set :deploy_via,     :copy
set :git_enable_submodules, true
set :copy_dir,       "./tmp"

set :copy_exclude, [
  ".git",
  ".gitignore",
  "/ild.xml",
  "/app/config/config_test.yml",
  "/bin",
  "/src/Application/MainBundle/Tests",
  "/src/Application/MainBundle/Resources/doc",
  "/web/apple-touch-icon.png",
```



```
]

# SETUP PROPER PERMISSION:

set :writable_dirs,      [ "app/cache", "app/logs" ]
set :webserver_user,     "www-data"
set :permission_method,  :acl
set :use_set_permissions, true

# COMPOSER SETTINGS:
set :use_composer,       true
#set :composer_bin,      "/home/usync/bin/composer"
set :interactive_mode,    false

# SYMFONY 2 SPECIFIC SETTINGS:

set :model_manager,      "doctrine" # Or: 'propel'
set :shared_files,       [
  "app/config/parameters.yml",
  "build.xml",
  "properties.cnf"
]
set :shared_children,     [
  log_path,
  web_path + "/uploads",
  "vendor"
]

set :update_vendors,      true
set :dump_assetic_assets, true
```

```
set :assets_symlinks,      true

# LOGGER VERBOSITY:

logger.level = Logger::MAX_LEVEL

# OTHER SETTINGS:

set :keep_releases, 3

# TASKS:

task :dbg do
  origin_file = "app_dev.php"
  destination_file = deploy_to + "/current/web/" + origin_file
  top.upload(origin_file, destination_file)
end

task :udbg do
  origin_file = "app_dev.php"
  destination_file = deploy_to + "/current/web/" + origin_file
  try_sudo "rm -rvf #{destination_file}"
end

task :fc do
  origin_file = "app.php"
  destination_file = deploy_to + "/current/web/" + origin_file
  try_sudo "mkdir -p #{File.dirname(destination_file)}"
  top.upload(origin_file, destination_file)
```

end

```
task :upload_addons do
  origin_file = "parameters.yml"
  destination_file = shared_path + "/app/config/#{origin_file}"
  try_sudo "mkdir -p #{File.dirname(destination_file)}"
  top.upload(origin_file, destination_file)

  origin_file = "properties.cnf"
  destination_file = shared_path + "/#{origin_file}"
  try_sudo "mkdir -p #{File.dirname(destination_file)}"
  top.upload(origin_file, destination_file)

  origin_file = "ant/build.xml"
  destination_file = shared_path + "/build.xml"
  try_sudo "mkdir -p #{File.dirname(destination_file)}"
  top.upload(origin_file, destination_file)
```

end

```
task :to_do_after_deploy do
  deploy.migrate
  deploy.cleanup
  upload_addons
  fc
```

end

```
# EVENT HANDLING:
```

```
before "deploy", "upload_addons"
```

```
after "deploy:setup", "upload_addons"
```

```
after "deploy", "to_do_after_deploy"
```

1.13 Narzędzia dodatkowe ant

Plik konfiguracyjny build.xml dla ant

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE project>

<project default="sync" basedir=". ">

  <target name="sync" depends="dep, cc"/>

  <target name="dep">
    <exec executable="cap">
      <arg line="deploy"/>
    </exec>
  </target>

  <target name="db">
    <exec executable="cap">
      <arg line="symfony:doctrine:database:drop"/>
    </exec>
    <exec executable="cap">
      <arg line="symfony:doctrine:database:create"/>
    </exec>
    <exec executable="cap">
      <arg line="symfony:doctrine:schema:update"/>
    </exec>
    <exec executable="cap">
```

```
        <arg line="symfony:doctrine:load_fixtures"/>
    </exec>
</target>

<target name="rb">
    <exec executable="cap">
        <arg line="deploy:rollback"/>
    </exec>
</target>

<target name="cc">
    <exec executable="cap">
        <arg line="symfony:cache:clear"/>
    </exec>
    <exec executable="cap">
        <arg line="symfony:cache:warmup"/>
    </exec>
    <exec executable="cap">
        <arg line="symfony:assets:install"/>
    </exec>
    <exec executable="cap">
        <arg line="symfony:assetic:dump"/>
    </exec>
    <exec executable="bash">
        <arg line="-c &quot;echo fos:js-routing:dump | cap symfony&quot;"/>
    </exec>
</target>

</project>
```

1.13.1 Przykład użycia

Wdrożenie zakończone czyszczeniem cache i wgraniem assets

```
$ ant
```

Reset bazy danych połączony z wgraniem fixtures.

```
$ ant db
```

Czyszczenie cache i wgranie assets.

```
$ ant cc
```

Przywrócenie poprzedniej wersji.

```
$ ant rb
```