Import settings from a bundle

> The following recipe is valid for any type of settings supported by Symfony framework.

Use case

Usually, you develop your website using one or several custom bundles as this is a best practice. However, dealing with core bundles semantic configuration can be a bit tedious if you maintain it in the main `ezpublish/config/ezpublish.yml` configuration file.

This recipe will show you how to import configuration from a bundle the manual way and the implicit way.

The manual way

This is the simplest way of doing and it has the advantage to be explicit. The idea is to use the `imports` statement in your main `ezpublish.yml` :

ezpublish/config/ezpublish.yml

```
imports:
    # Let's import our template selection rules that reside in our custom bundle.
    # MyCustomBundle is the actual bundle name
    - {resource: "@AcmeTestBundle/Resources/config/templates_rules.yml"}

ezpublish:
    # ...
```

templates_rules.yml, placed under Resources/config folder in AcmeTestBundle

```
# Here I need to reproduce the right configuration tree.
# It will be merged with the main one
ezpublish:
    system:
        my_siteaccess:
            ezpage:
                layouts:
                    2ZonesLayout1:
                        name: "2 zones (layout 1)"
                        template: "AcmeTestBundle:zone:2zoneslayout1.html.twig"

            location_view:
                full:
                    article_test:
                        template: "AcmeTestBundle:full:article_test.html.twig"
                        match:
                            Id\Location: [144,149]
                    another_test:
                        template: "::another_test.html.twig"
                        match:
                            Id\Content: 142

            block_view:
                campaign:
                    template: "AcmeTestBundle:block:campaign.html.twig"
                    match:
                        Type: "Campaign"
```

During the merge process, if the imported configuration files contain entries that are already defined in the main configuration file, **they will override them**.

Tip
If you want to import configuration for development use only, you can do so in your `ezpublish_dev.yml`

The implicit way

Compatible with eZ Publish 5.1 and higher versions / Symfony 2.2+ as it uses `Symfony\Component\DependencyInjection\Extension\PrependExtensionInterface` which is available as of Symfony 2.2 (more info on Symfony cookbook).

The following example will show you **how to implicitly load settings to be configure eZ Publish kernel**. Note that this is also valid for any bundle !

We assume here that you're aware of service container extensions.

Acme/TestBundle/DependencyInjection/AcmeTestExtension

```
<?php

namespace Acme\TestBundle\DependencyInjection;

use Symfony\Component\DependencyInjection\ContainerBuilder;
use Symfony\Component\Config\FileLocator;
use Symfony\Component\DependencyInjection\Extension\PrependExtensionInterface;
use Symfony\Component\HttpKernel\DependencyInjection\Extension;
use Symfony\Component\DependencyInjection\Loader;
use Symfony\Component\Yaml\Yaml;

/**
 * This is the class that loads and manages your bundle configuration
 *
 * To learn more see {@link
http://symfony.com/doc/current/cookbook/bundles/extension.html}
 */
class AcmeTestExtension extends Extension implements PrependExtensionInterface
{
    // ...

    /**
     * Allow an extension to prepend the extension configurations.
     * Here we will load our template selection rules.

     *
     * @param ContainerBuilder $container
     */
    public function prepend( ContainerBuilder $container )
    {
        // Loading our YAML file containing our template rules
        $config = Yaml::parse( __DIR__ . '/../Resources/config/template_rules.yml' );
        // We explicitly prepend loaded configuration for "ezpublish" namespace.
        // So it will be placed under the "ezpublish" configuration key, like in
ezpublish.yml.
        $container->prependExtensionConfig( 'ezpublish', $config );
    }
}
```

AcmeTestBundle/Resources/config/template_rules.yml

```
# We explicitly prepend config for "ezpublish" namespace in service container
extension,
# so no need to repeat it here
system:
    ezdemo_frontend_group:
        ezpage:
            layouts:
                2ZonesLayout1:
                    name: "2 zones (layout 1)"
                    template: "AcmeTestBundle:zone:2zoneslayout1.html.twig"

        location_view:
            full:
                article_test:
                    template: "AcmeTestBundle:full:article_test.html.twig"
                    match:
                        Id\Location: 144
                another_test:
                    template: "::another_test.html.twig"
                    match:
                        Id\Content: 142

        block_view:
            campaign:
                template: "AcmeTestBundle:block:campaign.html.twig"
                match:
                    Type: "Campaign"
```

Regarding performance
Service container extensions are called only when the container is being compiled, so there is nothing to worry about regarding performance.

In dev environment, each time you modify files loaded this way, **you'll need to manually clear** the cache as Symfony doesn't check modification time on them.

Loading configuration this way lets you override settings in your main `ezpublish.yml` file.