

# Using the Helper classes

## Using the helper classes

Helper classes enable you to generate standard HTML elements for the back office as well as for module configuration pages.

PrestaShop 1.5 introduced 5 Helper classes, along with the `Helper` parent class: `HelperForm`, `HelperOptions`, `HelperList`, `HelperView`, and `HelperHelpAccess`. Among these, only the first three will be useful to developer, the last two being more internal helpers for the first three.

PrestaShop 1.6 adds a handful more: `HelperCalendar`, `HelperUploader`, `HelperImageUploader`, `HelperKpi`, `HelperKpiRow`, `HelperTreeCategories`, and `HelperTreeShops`. Again of these have no direct use developers: their features are directly handled by the main Helpers. For instance, the new uploading system, handled by `HelperUpload` (and `HelperImageUploader` for product images), is directly used by `HelperForm` when generating a `<input type="file">` element.

Therefore, you will most of the time only deal with `HelperForm`.

## The Main Helpers

All the helper classes inherit from the `Helper` parent class:

- `HelperForm`: used to generate an edition form for an object of type `ObjectModel`. Example: editing the client's profile.
- `HelperOptions`: used to generate a configuration form, the values of which are stored in the `configuration` table. Example: the "Preferences" page.
- `HelperList`: used to generate a table of elements. The elements can belong to `ObjectModel`-type objects, but they do not have to. Example: client list, order status list, etc.

## The Helper Templates

The helpers use Smarty templates which are found in the following folder: `admin/themes/default/template/helpers/name_of_the_helper/`

Each template can be overloaded.

## Helper Overloading

An `AdminController` can overload any Helper template, simply by creating a `.tpl` file of the same name in the folder named `admin/themes/default/controllers/name_of_the_controller/helpers/name_of_the_helper/`

If possible, it should extend the parent template, not just replace it. Smarty 3 allows for inheritance by declaring `{block name=" "}` tags. A child template can overload a parent block by opening a block of the same name.

In addition to this section, you can read how to [use helpers to overload a back office template](#).

## Template inheritance example: adding a new type of field in a form

For the sake of this example, let's change the edition for the client's addresses. We want a field that would display the name and e-mail of the client, if these are known, or an e-mail input field otherwise.

We must create a new template: `/admin-dev/themes/default/template/controllers/addresses/helpers/form/form.tpl`

This template will contain the following code:

```
{extends file="helpers/form/form.tpl"}

{block name="field"}
    {if $input.type == 'text_customer'}
        {if isset($customer)}
            ...
        {else}
            ...
        {/if}
    {else}
        {$smarty.block.parent}
    {/if}
{/block}
```

We first declare the template's parent, then we can overload its `field` block. That block contains the field display. Our code checks the field type:

- If it is of type `text_customer`, then it handles the content display.
- If it is of any other type, it gives way to the parent's handling code.