Helpers

Helpers

PrestaShop 1.5 has Helper classes, which enable you to generate standard HTML elements for the back-office as well as for module configuration pages.

Several types of helper

There are 5 types of helper classes, all inheriting from the Help 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.
- HelperView: used to generate a View page. Example: the page that is displayed when the client lists her orders, her carts, etc.
- HelperHelpAccess: used to generate the toolbar's help link.

Their 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.

Overloading

An `AdminController` can overload any help 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 open 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 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.