Using the backward compatibility toolkit -

Using the backward compatibility toolkit

Description

PrestaShop's module API has greatly improved since version 1.4 of the software, and the API in version 1.5 and 1.6 is quite different.

Because there is a huge ecosystem of modules that are being upgraded to support the 1.5/1.6 module API while many shops are still using PrestaShop 1.4, the developers of PrestaShop chose to build Back ward Compatibility, a free toolkit which is only available for PrestaShop 1.4, and makes it possible to make 1.5/1.6 modules work in PrestaShop 1.4.

Including this toolkit makes it possible to develop a 1.4 module that uses PrestaShop 1.5/1.6 standards.

The 1.5/1.6 API's Helpers (HelperForm, HelperView, etc.) are still not available in PrestaShop 1.4.

Therefore, the following declarations are not necessary anymore:

```
global $smarty, $cookie;
```

The information they provide is available through these calls:

```
$this->context->customer->id;
$this->context->language->id;
$this->context->smarty->assign('content', 'empty');
...
```

Download and install

You can download it directly from PrestaShop's repository: https://github.com/PrestaShop/PrestaShop-backward_compatibility .
Just clone the Git project, then copy the `/backward_compatibility` folder to the root folder of the module you are developing. For instance, if your module is called TestModule, the folder should be here: `/modules/testmodule/backward_compatibility`.

The `/backward_compatibility` folder should contain the following files:

- `backward.ini`: the version number of the toolkit.
- `backward.php`: the main code.
- `Context.php`: adds a Context-like support to PrestaShop 1.4, as well as backward compatible Controller- and Customer-like methods.
- `Display.php`: enables the display of TPL files in the back office.
- `index.php`: just a file to prevent the display of the folder to visitors.

It is useless to install the toolkit in PrestaShop 1.5 or PrestaShop 1.6.

How to use the module

To properly us the module, you must first declare it in the module constructor method:

```
public function __construct()
{
    $this->name = 'testmodule';
    $this->tab = 'other';
    $this->version = '0.1';
    $this->author = 'Firstname Lastname';
    $this->need_instance = 0;

    parent::__construct();

    $this->displayName = $this->l('Test module');
    $this->description = $this->l('This is a test module');
    $this->confirmUninstall = $this->l('Warning: all the data saved in your database will be deleted. Are you
sure you want uninstall this module?');

    /* Backward compatibility */
    if (_PS_VERSION_ < '1.5')
            require(_PS_MODULE_DIR_.$this->name.'/backward_compatibility/backward.php');
}
```

Once this is in place, you can use more of the 1.5/1.6 API's goodness, such as the Context.