

```
<?php
/**
 * Smarty Templating for Zend Framework
 *
 * Easily utilize the Smarty Templating Engine inside
 * the Zend Framework without hassle. Programmers
 * interested in Smarty are encouraged to replace
 * Zend_View with Zend_Smarty. Instantly, Zend_Smarty
 * should handle Zend_View's API allowing non-Smarty
 * programmers or existing projects fast access to the
 * entire benefits of the Smarty Templating Engine.
 *
 * @name      Zend_Smarty
 * @author    David Branco <David@NeoliteUSA.com>
 * @link      http://www.NeoliteUSA.com/
 * @version   0.7.6
 * @since     Jan 04, 2007
 * @license   http://creativecommons.org/licenses/by/2.5/
 */

require_once('./libs/Smarty/Smarty.class.php');

class Zend_Smarty extends Smarty
{
    /**
     * Holds the root path of your web application
     * @var string
     */
    public $rootPath;

    /**
     * Assigned variables.
     * @var array
     */
    private $_vars = array();

    public function __construct($rootPath)
    {
        $this->root_path = $rootPath;
        $this->Smarty();
        $this->debugging = false;
        $this->template_dir = $this->rootPath . 'views/';
        $this->compile_dir = $this->rootPath . 'cache/tplcache/';
        $this->config_dir = $this->rootPath . 'cache/configs/';
        $this->cache_dir = $this->rootPath . 'cache/cache/';
        $this->plugins_dir[] = 'extra_plugins';
        $this->caching = false;
        $this->StartPagination();
        return true;
    }

    /**
     * Directly assigns a variable to the view script.
     *
     * Note that variable names may not be prefixed with '_'.
     *
     * @param string $key The variable name.
     * @param mixed $val The variable value.
     * @return void
     */
    public function __set($key, $val)
    {
        /**
         * @todo exception?
         */
        if ($key[0] != '_') {
```

```

        $this->_vars[$key] = $val;
        $this->assign($key, $val);
    }
}

/**
 * Retrieves an assigned variable.
 *
 * Note that variable names may not be prefixed with '_'.
 *
 * @param string $key The variable name.
 * @return mixed The variable value.
 */
public function __get($key)
{
    /**
     * @todo exception?
     */
    if ($key[0] != '_') {
        return isset($this->_vars[$key]) ? $this->_vars[$key] : null;
    }
}

/**
 * Allows testing with empty() and isset() to work inside
 * templates -- only available on PHP 5.1
 *
 * @param string $key
 * @return boolean
 */
public function __isset($key)
{
    return array_key_exists($key, $this->_vars) && ($key[0] != '_');
}

function render($template_name) {
    $this->assign('urlPrefix', $GLOBALS['config']['urlPrefix']);
    $this->display($template_name . '.tpl');
    exit();
}

function StartPagination() {
    require_once($this->root_path . 'helpers/SmartyPaginate.php');
}

function AssignWithPagination($tpl_var, &$value, $item_limit=25, $page_limit=10) {
    $page_limit = $page_limit-4;
    $tpl = Zend::registry('view');
    $filterGet = Zend::registry('FilterGet');
    if (!empty($tpl_var)) {
        if(!SmartyPaginate::isConnected($tpl_var)) {
            SmartyPaginate::connect($tpl_var);
            SmartyPaginate::setUrlVar($tpl_var.'_next',$tpl_var);
            SmartyPaginate::setLimit($item_limit,$tpl_var);
            SmartyPaginate::setPageLimit($page_limit,$tpl_var);
        }else{
            SmartyPaginate::connect($tpl_var);
        }
        $values = $value;
        SmartyPaginate::setTotal(count($values),$tpl_var);
        $values = array_slice($values, SmartyPaginate::getCurrentIndex($tpl_var),
SmartyPaginate::getLimit($tpl_var));
        parent::assign_by_ref($tpl_var, $values);
        SmartyPaginate::assign($tpl, 'paginate', $tpl_var);
    }
}

```

```
}
```