

ClickBlocks\Web\POM\Panel

General information

Inheritance	ClickBlocks\Web\POM\Control
Subclasses	All classes of container controls.
Interfaces	ArrayAccess, IteratorAggregate, Countable
Source	Framework/web/pom/controls/panel.php

Panel is the base class for all web controls that can contain other controls (such controls are called container controls). The class provide the common functionality of a container control:

- adding, removing and replacing child controls of the panel.
- search child controls within the panel.
- caching of the panel template and setting new template.

Besides of properties that common for all controls, **Panel** have the following properties that common for all container controls:

Name	Type	Default value	Description
tag	string	div	the HTML tag name of the main element of a container control.
expire	integer	0	the expiration time (in seconds) of the panel template cache.

Since **Panel** implements interface **IteratorAggregate** it is possible to iterate all child controls of the panel in foreach cycle:

```
foreach ($panel as $UID => $ctrl)
{
    echo 'Child control [UniqueID: ' . $UID . ', LogicID: ' . $ctrl['id'] . ']';
}
```

Public non-static properties

tpl

```
public ClickBlocks\Core\Template $tpl
```

The panel template object.

Public non-static methods

__construct()

```
public void __construct(string $id, string $template = null, integer $expire = 0)
```

\$id	string	the logic identifier of the control.
\$template	string	the panel template or the full path to the template file.
\$expire	integer	the cache lifetime (in seconds) of the panel template.

Class constructor. Initializes the panel template object.

getVS()

```
public array getVS()
```

Returns view state of the panel.

setVS()

```
public self setVS(array $vs)
```

\$vs	array	the panel view state.
-------------	-------	-----------------------

Sets view state of the panel.

getControls()

```
public array getControls()
```

Returns the child controls of the panel.

setControls()

```
public self setControls(array $controls)
```

\$controls	array	the child controls of the panel.
-------------------	-------	----------------------------------

Sets child controls of the panel.

count()

```
public integer count()
```

Returns number of the panel controls.

getIterator()

```
public ClickBlocks\Web\POM\Iterator getIterator()
```

Returns the control iterator object.

parse()

```
public self parse(string $template = null, array $vars = null)
```

\$template	string	the template string or full path to the template file.
\$vars	array	the template variables for the template preprocessing.

Parses the given template of the panel. If **\$template** is not defined, the current panel template will be parsed.

add()

```
public self add(ClickBlocks\Web\POM\Control $ctrl, string $mode = null, string $id = null)
```

\$ctrl	ClickBlocks\Web\POM\Control	any control to be added to the panel.
---------------	-----------------------------	---------------------------------------

\$mode	string	determines the placement of the adding control in the panel template.
\$id	string	the unique or logic identifier of some panel control or value of the attribute "id" of some element in the panel template.

Adds some control on the panel.

This method is analogue of method **setParent()** of class **Control**.

detach()

```
public self detach(string $id)
```

\$id	string	the unique or logic identifier of the control.
-------------	--------	--

Removes the given control from the panel.

replace()

```
public self replace(string $id, ClickBlocks\Web\POM\Control $new)
```

\$id	string	the unique or logic identifier of the panel control to be replaced.
\$new	ClickBlocks\Web\POM\Control	the control to replace.

Replaces some panel control with the given control.

copy()

```
public ClickBlocks\Web\POM\Panel copy(string $id = null)
```

\$id	string	the logic identifier of the control copy.
-------------	--------	---

Creates full copy of the control. If **\$id** is not defined the logic identifier of the original control is used.

check()

```
public self check(boolean $flag = true, boolean $searchRecursively = true)
```

\$flag	boolean	determines whether a checkbox will be checked or not.
\$searchRecursively	boolean	determines whether the method should be recursively applied to all child panels of the given panel.

Checks or unchecks checkboxes of the panel.

clean()

```
public self clean(boolean $searchRecursively = true)
```

\$searchRecursively	boolean	determines whether the method should be recursively applied to all child panels of the given panel.
----------------------------	---------	---

For every panel control restores value of the property "value" to default.

get()

```
public boolean|ClickBlocks\Web\POM\Control get(string $id, boolean $searchRecursively = true)
```

\$id	string	the unique or logic identifier of a control of the panel.
\$searchRecursively	boolean	determines whether to recursively search the control inside all child panels of the given panel.

Searches the required control in the panel and returns its instance. The method returns FALSE if the required control is not found.

getFormValues()

```
public array getFormValues(boolean $searchRecursively = true)
```

\$searchRecursively	boolean	determines whether values of controls of nested panels will also be gathered.
----------------------------	---------	---

Returns associative array of values of form (panel) controls. The keys of this array are logic identifiers of the panel controls and its values are values of property **value** of the panel controls.

setFormValues()

```
public self setFormValues(array $values, boolean $searchRecursively = true)
```

\$values	array	values to be assigned to.
\$searchRecursively	boolean	determines whether values will be also assigned to the controls of nested panels.

Assigns values to the panel controls. The keys of values array should be logic identifiers of the panel controls and its values should be values of property **value** of the panel controls.

render()

```
public string render()
```

Returns HTML of the panel.

renderInnerHTML()

```
public string renderInnerHTML()
```

Returns the inner HTML of the panel.

renderXHTML()

```
public string renderXHTML()
```

Returns XHTML of the control.

Protected non-static properties

ctrl

```
protected $ctrl = 'panel'
```

The control type.

controls

```
protected array $controls = []
```

inDetach

```
protected $inDetach = false
```

It equals TRUE if some control detaches from the panel. Otherwise it is FALSE.