YUI Library: Panel

Simple Use Case: YAHOO.widget.Panel

Markup (optional, using standard module format):
```html
<div id="myPanel">
    <div class="hd">Header content.</div>
    <div class="bd">Body content.</div>
    <div class="ft">Footer content.</div>
</div>
```

Script:
```javascript
var oPanel = new YAHOO.widget.Panel("myPanel");
oPanel.render();
oPanel.show();
```

Creates, renders and shows a panel using existing markup and all default Panel settings.

Constructor: YAHOO.widget.Panel

```javascript
YAHOO.widget.Panel(str elId[, obj config]);
```

Arguments:
(1) **Element ID:** HTML ID of the element being used to create the Panel. If this element doesn't exist, it will be created.
(2) **Configuration Object:** JS object defining configuration properties for the panel. See Configuration section for full list.

Solutions

There are three ways to **configure options on your Panel**:

```javascript
// 1. In the constructor, via an object literal:
var myPanel = new YAHOO.widget.Panel("myPanel", {
    visible:false });
// 2. Via "queueProperty", prior to rendering:
myPanel.cfg.queueProperty("visible",false);
// 3. Via "setProperty" after rendering:
myPanel.cfg.setProperty("visible",false);
```

Align the top left corner of your Panel with the bottom right corner of an element whose HTML ID is "contextEl":

```javascript
myPanel.cfg.setProperty("context", ["contextEl",
    "tl", "br"]);
```

Subscribe to a Panel Custom Event, listening for changes to the Panel's postion, alerting its new position after move:

```javascript
alertMove = function(type, args) {
    alert(args[0] + ", " + args[1]);
}
myPanel.subscribe("move", alertMove);
```

Key Interesting Moments in Panel

See online docs for a complete list of Panel's Custom Events.

Event	Arguments
beforeRenderEvent	None.
renderEvent	None.
beforeShowEvent	None.
showEvent	None.
beforeHideEvent	None.
hideEvent	None.
beforeMoveEvent	X, Y to which the Panel will be moved.
moveEvent	X, Y to which the Panel was moved.
hideMaskEvent	None.
showMaskEvent	None.
changeContentEvent	None.
changeBodyEvent	String or element representing new body content (**Note:** there are corresponding Header and Footer change events, too).

All Panel events are YUI Custom Events (see Event Utility docs); subscribe to these events using their subscribe method: `myPanel.hideEvent.subscribe(fnMyHandler);`.

Key Panel Configuration Options

See online docs for complete list of Panel options; see Solutions (bottom left) for how to set your options.

Option (type)	Default	Description
close (b)	null	Display close icon.
draggable (b)	null	Make the Panel draggable.
modal (b)	null	Use a modal mask behind Panel when Panel is visible.
visible (b)	true	Sets the "display" style property to "block" (true) or "none" (false).
x, y, and xy (int, int, ar)	null	These properties can be used to set the Panel's "top" and/or "left" styles.
context (ar)	null	Anchors Panel to a context element; format: `[el contextEl, s panelCorner, s contextCorner]` with corners defined as "tr" for "top right" and so on.
fixedcenter (b)	false	Automatically center Panel in viewport?
width (s)	null	Sets "width" style property.
height (s)	null	Sets "height" style property.
zindex (int)	null	Sets "z-index" style property.
constrainto viewport (b)	false	When true, prevents the Panel from being dragged out of the viewport.
underlay (s)	"shadow"	Type of underlay: "shadow", "none", or "matte".
effect (obj)	null	Object defining effect (FADE or SLIDE) to use in showing and hiding Panel: `{effect: YAHOO.widget.ContainerEffect.FADE, duration:1}`

YAHOO.widget.Panel: Properties

body (el)
element (el) containing header, body & footer
footer (el)
header (el)
id (s) of the element

YAHOO.widget.Panel: Methods

appendToBody(el *element*)
appendToFooter(el *element*)
appendToHeader(el *element*)
hide()
render([el *element*])
Argument required for Panels not built from existing markup. Panel will not be in the DOM or visible until render is called
setBody(str or el *content*)
setFooter(str or el *content*)
setHeader(str or el *content*)
show()
bringToTop()

Dependencies

Panel requires the full Container package, the YAHOO object, Event, and Dom. Animation, and Drag and Drop are optional. **Note:** Panels use Drag and Drop by default — a simple Panel with default configuration options will not be draggable without it.