

Simple Use Case: YAHOO.util.Resize

Markup:

```
<div id="resizeMe"><p>Lorem ipsum</p></div>
```

Script:

```
var resize = new YAHOO.util.Resize('resizeMe');
```

Creates a Resize instance with default configurations.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse justo nibh, pharetra at, adipiscing ullamcorper, rutrum ac, enim. Nullam pretium interdum metus. Ut in neque. Vivamus ut lorem vitae turpis porttitor tempor.

Constructor: YAHOO.util.Resize

```
YAHOO.util.Resize(str | el ref container[, obj config])
```

Arguments:

- (1) **Element:** The element to make resizable.
- (2) **Configuration object:** When instantiating Resize, you can pass all configurations in as an object argument or configure the instance after instantiation. See Configuration Options section for common configuration object members.

Solutions: Customizing the Proxy Element

The following code demonstrates customization of the proxy element.

```
//instantiate Resize:
var myResize = new YAHOO.util.Resize('resizeMe', {
    proxy: true
});

//customize proxy during resize via innerHTML:
myResize.on('startResize', function() {
    myResize.getProxyEl().innerHTML = 'I am the proxy';
    YAHOO.util.Dom.setStyle(myResize.getProxyEl(),
        'opacity', '5');
});
```

Resize Events

Event	Fires...
dragEvent	Fires when the DragDrop dragEvent is fired for the config option draggable.
startResize	Fires when a resize action is started.
beforeResize	Fires before every element resize, after the size calculations have been done. Returning false will cancel the resize.
resize	Fires on event element resize (only fires once when used with proxy config setting)
proxyResize	Fires on every element resize (only fires when used with proxy config setting).

All Resize events are Custom Events (see Element docs); subscribe to these events using their subscribe method: `resize.on('resize,fnMyHandler');`.

Key Resize Configuration Options

Option	Default	Description
proxy	false	Resize a proxy element instead of the actual element.
animate	false	Indicates whether or not the resize should animate sizes (only works with proxy).
status	false	Should we show the status tooltip.
handles	['r', 'br', 'b']	The handles to use (any combination of): 't', 'b', 'r', 'l', 'bl', 'br', 'tl', 'tr'. Can use a shortcut of All. Note: 8 way resizing should be done on an element that is absolutely positioned.
ratio	false	Constrain the resize to a ratio of the current size.
draggable	false	A convenience method to make the element draggable.

Most of Resize options can be set in the constructor's second argument (eg, `{animate: true}`) or at runtime via `set` (eg, `resize.set("animate", true);`).

Resize Handles

The Resize Utility supports the following handle positions: Top, Bottom, Left, Right, Top Right, Top Left, Bottom Right, Bottom Left. The default handle positions are: Right, Bottom, and Bottom Right.

The default look of the handles is to take up all available space around the element to be resized. There are a few configuration options built in that will alter this look:

- hiddenHandles** - Handles are always transparent, the user gets feedback from the cursor change.
- hover** - Handles are hidden by default until the user hovers over them, then they appear.
- knobHandles** - Used for the classic 8-way resize.

Take a look at the Resize Utility's examples for demos of all of these options.

Note: To get the best effect out of using all 8 resize handles, it is recommended that the element be absolutely positioned (and if possible be a direct child of the body).

YAHOO.util.Resize: Methods

- getActiveHandleEl()** Get the HTML reference for the currently active resize handle.
- getProxyEl()** Get the HTML reference for the proxy, returns null if no proxy.
- getResizeById()** Get a resize object by the HTML id of the element associated with the Resize object.
- getStatusEl()** Get the HTML reference for the status element.
- getWrapEl()** Get the HTML reference for the wrap element, returns the current element if not wrapped.
- isActive()** Returns true or false if a resize operation is currently active on the element.
- reset()** Resets the element to its start state.
- resize()** Resizes the element, wrapper, or proxy based on the data from the handlers.
- destroy()** Destroys the resize object and all of its elements & listeners.

Dependencies

The Resize Utility requires Yahoo, Dom, Element, Event, Drag and Drop and Animation (optional).