

# Y! YUI Library: ImageCropper Control [beta]

2009-2-6

v2.7

## Simple Use Case: YAHOO.widget.ImageCropper

### Markup:

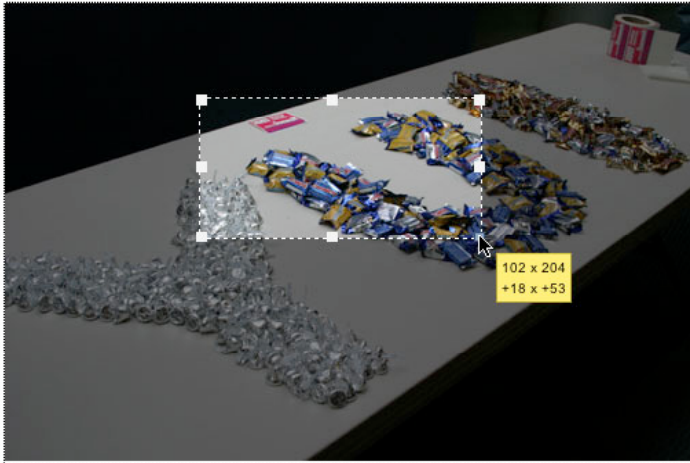
```

```

### Script:

```
crop = new YAHOO.widget.ImageCropper('crop1');
```

Using the above code will get you a rich control that looks something like this:



## Constructor: YAHOO.widget.ImageCropper

```
YAHOO.widget.ImageCropper(str | el ref  
  container[, obj config])
```

### Arguments:

- (1) **Element:** The element to make croppable.
- (2) **Configuration object:** When instantiating the ImageCropper Control, 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.

## Dependencies

ImageCropper requires the Yahoo Global Object, Dom, Event, Element, DragDrop, and Resize.

## ImageCropper Events

See online docs for complete list of Resize events.

Event	Fires...
dragEvent	Fires when the DragDrop dragEvent is fired.
startResizeEvent	Fires when a resize action is started.
resizeEvent	Fires on event element resize
moveEvent	Fires on every element move. Inside these methods: <code>_handleKeyPress</code> , <code>_handleDragEvent</code> , <code>handleResizeEvent</code>

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

## Key ImageCropper Configuration Options

See online docs for complete list of ImageCropper configuration options.

Option	Default	Description
initHeight	1/4 of the image height	Set the initial height of the crop area.
initWidth	1/4 of image width	Set the initial width of the crop area.
initialXY	[10,10]	Array of the XY position that we need to set the crop element to when we build it.
keyTick	1	The pixel tick for the arrow keys.
ratio	false	Constrain the resize to a ratio of the current size.
status	True	Show the Resize Utility status.

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

## Getting the crop area

```
//set up a new cropper  
var cropImg = new YAHOO.widget.ImageCropper('img1');  
//get the crop area  
var cropArea = cropImg.getCropCoords();  
  
// cropArea now looks like this:  
  
{  
  top: 70,  
  left: 122,  
  height: 86,  
  width: 172,  
  image: 'yui.jpg'  
}
```

## YAHOO.widget.ImageCropper: Methods

- getActiveHandleEl()** Get the HTML reference for the currently active resize handle.
- getCropCoords()** Get the coordinates needed to crop the image
- getCropperById()** Get an ImageCropper object by the HTML id of the image associated with the ImageCropper object.
- getEl()** Get the HTML reference for the image element.
- getMaskEl()** Get the HTML reference for the mask element.
- getResizeEl()** Get the HTML reference for the resize element.
- getResizeMaskEl()** Get the HTML reference for the resizable object's mask element.
- getResizeObject()** Get the Resize Utility object.
- getWrapEl()** Get the HTML reference for the wrap element.
- destroy()** Destroys the ImageCropper object and all of its elements & listeners.