

HTML5 drawing markup control

[Technologies](#) > [User Interface Add-Ins](#) > [HTML5 Markup Control](#) > [Markup Control with Floor Plan](#)

The Markup Control enables users to mark up images, floor plans, and other SVG drawings with:

- redlines
- legends of highlight colors
- room highlights for a plan type, such as vacant rooms (floor plans only)
- room highlights for a filter you specify or a set of rooms that you select (floor plans only)

To use all of the Markup Control's tools, specify the ab-common-controls-svg-markup-drawing.axvw file:

```
<panel type="view" id="markupDrawingControl" file="ab-common-controls-svg-markup-drawing.axvw"/>
```

When you specify this file, the view contains all sub-controls, CSS, localized strings, and third-party libraries.

To use a simple, color-selectable pen for markup, specify the ab-common-controls-svg-drawing.axvw file:

```
<panel type="view" id="drawingControlWithRedlining" file="ab-common-controls-svg-drawing.axvw"/>
```

When you specify this file, the view does not include plan type highlight, filter highlight, or room selector sub-controls.

Alternately, you can customize your markup tools to include exactly what you need. In that case, install the [minimum required libraries](#) below, then add [optional libraries](#) as required.

Minimum required libraries

DrawingSvgService

```
<js url="/archibus/dwr/interface/DrawingSvgService.js" />
```

d3 SVG library

```
<js file="d3.min.js"/>
```

Drawing control and markup drawing control

```
<js file="ab-svg-drawing.js"/>
```

```
<js file="ab-svg-drawing-control.js"/>
```

```
<js file="ab-svg-markup-drawing.js"/> (if markup drawing control is needed)
```

Image control

```
<js file="ab-svg-image.js"/> (if markup drawing control is needed)
```

Upload or save action control

```
<js file="ab-svg-actions.js"/> (if markup drawing control is needed)
```

Common functions used by markup control

```
<js file="ab-svg-common.js"/>
```

Optional libraries

To add redlining legend control:

- CSS files:

```
<css file="redline.css"/>
```

- Redline control:

```
<js file="redline.js"/>
```

```
<js file="ab-svg-redline.js"/>
```

```
<js file="ab-svg-redline-control.js"/>
```

- Color picker control:

```
<js file="jscolor.js"/>
```

```
<js file="ab-svg-redline-colorpicker.js"/>
```

To add placement control:

- CSS files:

```
<css file="placement.css"/>
```

- Placement control:

```
<js file="placement.js"/>
```

```
<js file="ab-svg-placement.js"/>
```

```
<js file="ab-svg-placement-control.js"/>
```

To add plan type highlight control:

```
<js file="ab-svg-planypehighlight-control.js"/>
```

To add filter highlight control, add:

```
<js file="ab-svg-filterhighlight-control.js"/>
```

```
<js file="ab-svg-roomselector-control.js"/>
```

Initialization

Method	Description
constructor: function(DivId, panelId, config)	<p>Where:</p> <p>divId - String Id of <div/> that holds the svg.</p> <p>panelId - String Id of the panel that holds svg drawing.</p> <p>config - an Ab.view.ConfigObject object that contains the ids of sub controls and actions.</p> <p>You can pass in these parameters:</p> <ul style="list-style-type: none">• planTypeGroup – plan type group value from plantype_groups.plantype_group field. Used to propagate the plan type highlight list of selection items.• svgActions – the svg actions. By default, there are two actions: 'uploadImage' and 'saveSvg'. You can pass in one, two, or none of these values.• redlineLegend – the redlining legend control's panel id, div id and color picker div id, if any.• planTypeHighlight – the plan type highlight control's panel id and div id.

Initialize **MarkupDrawingControl** once, and only once, within the view controller's **AfterViewLoad API**. Refer to the sample code below:

```
var parameters = new Ab.view.ConfigObject();
parameters['planTypeGroup'] = 'Space Console';
parameters['svgActions'] = {uploadImage: 'uploadImageAction',
                           saveSvg: 'saveSvgAction'
                          };
parameters['redlineLegend'] = {panelId: 'redlineLegendPanel', divId: 'redlineLegendDiv', colorPickerId: 'redlineLegendPanel_head'};
this.MarkupDrawingControl = new Ab.svg.MarkupDrawingControl(divId, panelId, parameters);
```

Load SVG Drawing with sub controls

When the Building Code, Floor Code, or Action Item changes, an SVG drawing is loaded.

Method	Description
--------	-------------

<p>load: function(svgDivId, parameters, eventHandlers)</p>	<p>Where:</p> <p>svgDivId - String Id of <div/> that holds the svg drawing.</p> <p>parameters - Object. Server-side required information, such as as bl_id, fl_id, viewName, highlight and label dataSource names, to get and process corresponding SVG.</p> <p>The parameters can include existing parameters of constructor, if you want to overwrite. In addition, you must provide:</p> <ul style="list-style-type: none"> • pkeyValues - primary key values for floor plan, such as bl_id, fl_id. • activityLogId - action item id. <p>eventHandlers - Array of event handlers.</p>
--	--

Here is an example:

```

var parameters = new Ab.view.ConfigObject();
parameters['pkeyValues'] = {'bl_id':bl_id, 'fl_id':fl_id};
parameters['activityLogId'] = activity_log_id;
// load the floorplan
this.MarkupDrawingControl.load("drawingDiv", parameters);

```

Other API for Ab.svg.MarkupDrawingControl

Method	Description
enableActions: function(show, hasFloorPlan)	<p>Enable or disable Upload Image and Save panel actions. The actions also show or hide based on if it is floor plan, or if there are any unsaved changes. If there is a floor plan attached to the current action item, the "Upload Image" action will always be disabled.</p> <p>If there are no changes to the SVG drawing, the "Save" action will be disabled.</p> <p>Where:</p> <p>show boolean true to show actions, and false to hide.</p> <p>hasFloorPlan boolean true if there is a floor plan attached to the action item, false otherwise.</p>
showAction: function(actionId, show)	<p>Enable or disable the specified action.</p> <p>Where:</p> <p>actionId String - the action id to show or hide. Possible values includes all default actions: "filterRoomsAction", "selectRoomsAction", "uploadImageAction", "saveSvgAction" and any other user-defined panel actions.</p> <p>show boolean true to show the specified action, and false to hide.</p>
addRedmarks: function(redmarks)	<p>Add the specified redmarks to the svg drawing.</p> <p>Where:</p> <p>redmarks - String. A string that contains all redline elements from svg drawing.</p>
showLegend: function(show)	<p>Show or hide redline legend, upload control panel and svg drawing</p> <p>Where:</p> <p>show boolean true to show legend panel, svg drawing panel and hide upload image panel; false to hide legend, svg drawing panels and show upload image panel.</p>
loadReadlineControl: function()	<p>Create and load redline legend control with color picker, attach onDrop event handler.</p>
showPlanTypeHighlight: function(show)	<p>Show or hide background highlight control panel. If the background highlight control has never been created, create the control with config parameters passed from client side.</p> <p>Where:</p> <p>show boolean true to show plan type highlight selection panel, false to hide it.</p>

showFilterHighlight: function(show)	<p>Show or hide filter highlight control panel. If the filter highlight control has never been created, create the control with config parameters passed from client side.</p> <p>Where:</p> <p>show boolean true to show filter highlight selection panel, false to hide it.</p>
clearDrawing: function()	<p>Clear SVG drawing content.</p> <p>Save SVG drawing's extent, drawing name, action item id to afm_redlines table.</p> <p>Save redlines and image content (if any) to its document table, and update the related file name in afm_redlines table.</p>
save: function()	<p>Save highlight filter to its document table, and update the related file name in afm_redlines table.</p> <p>Capture the floor plan or image with redlines and highlight, save to activity_log.doc4 field.</p>

New Event handler for Ab.svg.MarkupDrawingControl

Method	Description
onRoomClicked: function(assetId, svg, control)	<p>Event handler to add highlight and border to the room asset.</p> <p>Where:</p> <p>assetId - String. id for room asset, i.e. HQ;17;101</p> <p>svg - DrawingSvg drawing content.</p> <p>control - Ab.svg.MarkupDrawingControl the control</p>

For example:

```
self.addEventHandlers(this, [{'assetType' : 'rm',
                             'handler' : this.onRoomClicked.createDelegate(this, [control], 2),
                             'highlightOnly': false}]);
```

Copyright © 1984-2015, ARCHIBUS, Inc. All rights reserved.