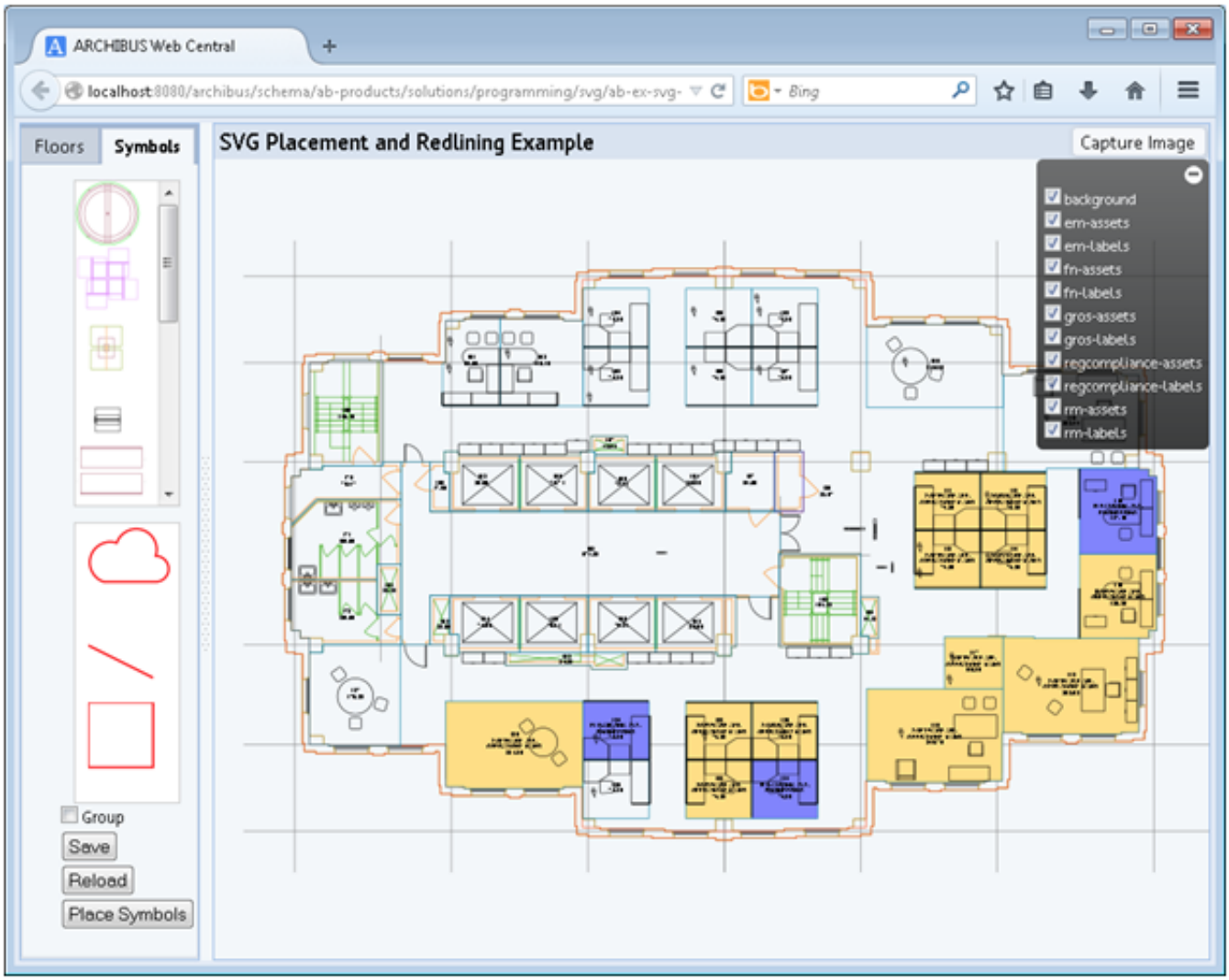
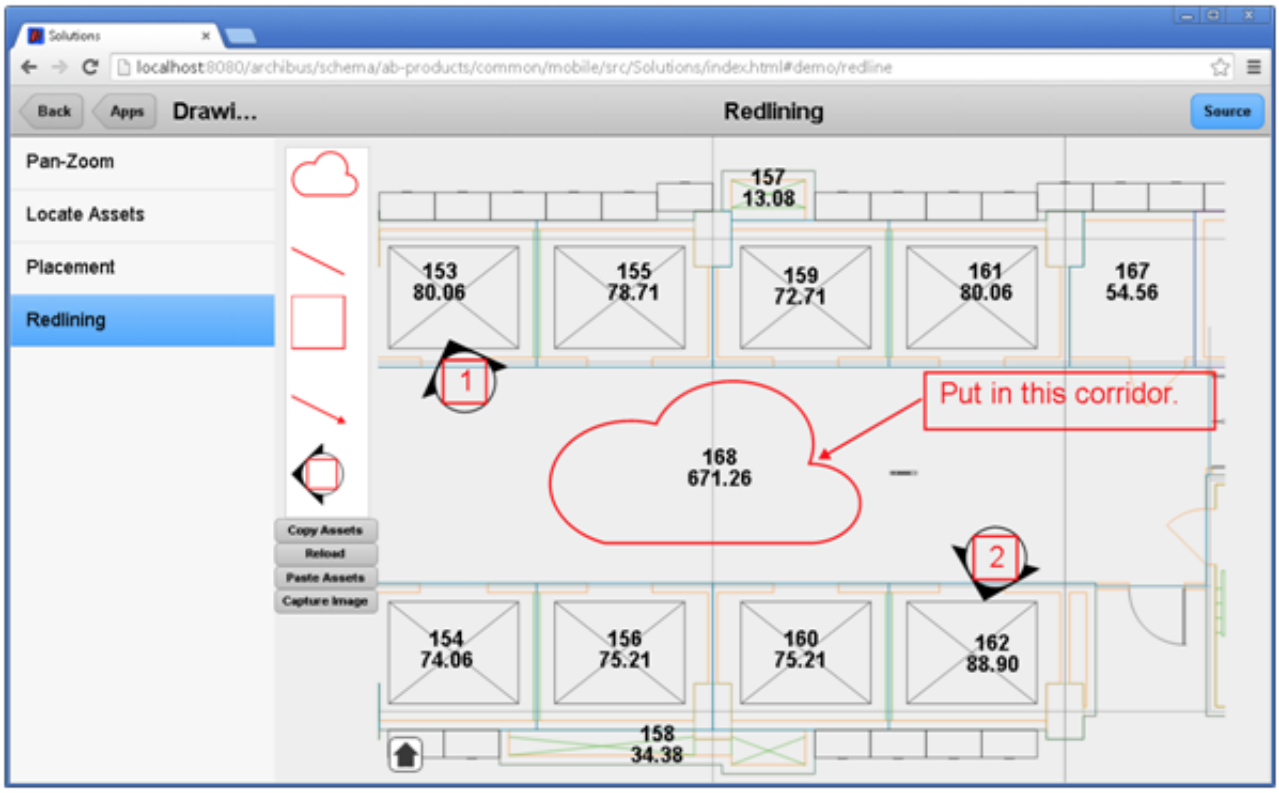
Add redlines to an HTML5 Drawing

You can provide Web Central and mobile users with the ability to mark up HTML5 drawings, such as adding notes, highlighting details, or otherwise indicating needed changes. This is known as redlining.



Note that in Web Central both redline features and placement legends can exist in the same view and you have flexibility in their location. In the above example, the legend actually exists in an entirely separate panel than the drawing.

Here is an example for mobile:



The Web Central and mobile versions of the control contain relatively minor syntax differences at the API level. However, both versions follow the same fundamental approach and setup. The following section walks through the major steps for the Web Central setup, with notes relevant to mobile.

Setting up a view for redlining

1. In the .axvw, add calls to the necessary .js and .css files, such as:
   ```
   <css file="redline.css"/>
   <js file="ab-svg-redline.js"/>
   <js file="ab-svg-redline-control.js"/>
   ```
   For mobile: require `Common.control.Redline` and `Common.control.Drawing`.

2. Add a div for the legend. Ensure that `class="redline-legend"`.
   ```
   <div id="redlineLegend" class="redline-legend" style="height: 250px; overflow-x:hidden"></div>
   ```
   For mobile: Steps 2 and 5 can be combined, as such:

   ```
   {
     itemId: 'redlineLegendPanel',     // container id
     xtype: 'redline',                 // xtype = redline
     legendId: 'redlineLegend',       // id of the legend's div
     drawingId: 'redlineDrawing',     // id of the drawing's div
     height: '360px',
     html: '<div id="redlineLegend" class="redline-legend"></div>'
   },
   ```

3. Add a div for the drawing.
   ```
   <div id="drawingId"></div>
   ```
   For mobile: this is:

   ```
   {
     itemId: 'redlineLegendPanel',     // container id
     xtype: 'redline',                 // xtype = redline
     legendId: 'redlineLegend',        // id of the legend's div
     drawingId: 'redlineDrawing',      // id of the drawing's div
   ```

```
  html: '<div id="redlineLegend" class="redline-legend"></div>'
},
```

4. Follow the regular steps to load the drawing into the drawing div. For Web Central, use the `setControlPanel` method to keep the control in sync with the window size.

```
// resize specified DOM element whenever the panel size changes
this.drawingPanel.setContentPanel(Ext.get('panelId'));
this.drawingPanel.setContentPanel(Ext.get('drawingId'));
```

5. In the .js file, create an instance of the control:.

```
this.redlineControl = new Ab.svg.RedlineControl(drawingId, panelId, legendId, parameters);
```

   for example:

```
this.redlineControl = new Ab.svg.RedlineControl("drawingId", "panelId", "legend", parameters);
```

6. In the .js file, load the legend via `loadLegend()`, per below. `loadLegend()` will establish the connection between the legend and the drawing. Redlines will be dropped to the “redlines” layer.

```
this.redlineControl.loadLegend({});
```

Ab.svg.RedlineControl API

Method	Description
loadLegend	Loads the legend. Setup is automatically used as the callback. All redlines will be dropped onto the ‘redlines’ layer. Parameters: • `config` Object. Object of additional options. Currently, only used as a placeholder for future needs.
setup	Establish the drag and drop relationship between the redline legend and the drawing. Parameters: None
copyAssets	Each asset dropped onto the drawing will be tagged with a “dropped” class. This method queries for all nodes under the “redlines” layer with for this class. Returns: • Array of redline `SVGGElements` dropped onto the drawing
pasteAssets	Places an array of `SVGGElements` /symbols onto the drawing. Hide interactive grips. Parameters: • `assetsToPaste` – Array. Array of `SVGGElements` to paste onto the drawing Returns: • Array of `SVGGElements` pasted on the drawing
getControl	Returns the redline control.
addRedmarks: function	Add the specified redmarks to the SVG drawing. Where:

(redmarks)	redmarks - String. A string that contains all redline elements from SVG drawing.
show: function(show)	Show or hide redline legend control panel. Where: show boolean true to show redline legend control panel, false otherwise.
setCurrentColor: function(newColor)	Set the redline control legends to a new color. Where: newColor - Color to set in HTML color code format, such as #FF0000. **Note:** Inherited from RedlineSvg.
retrieveRedmarks: function()	Return the SVG as a string. Include styles when necessary. Where: returns {string} - A string of redlines SVG elements. **Note:** Inherited from RedlineSvg.
attachRedlineEvents	Attach click event the existing redlines. **Note:** Inherited from RedlineSvg.

Sample views

Web Central

- http://localhost:8080/archibus/schema/ab-products/solutions/programming/svg/ab-ex-svg-dwg-placement-and-redlining.axvw

Mobile

- http://localhost:8080/archibus/schema/ab-products/common/mobile/src/Solutions/index.html#demo/redline