Interactive Svg Map Component version 6.0

Syntax

interactiveSvgMapField (label, labelPosition, instructions, helpTooltip, required, disabled, validations, height, showWhen, mapSvg, mapAreaParameters, hoverEffect, selectedIndex)

- **label** (Text): Text to display as the field label.
- **labelPosition** (Text): Determines where the label appears. Valid values: "ABOVE" (default), "ADJACENT", "JUSTIFIED", "COLLAPSED".
- **instructions** (Text): Supplemental text about this field.
- **helpTooltip** (Text): Displays a help icon with the specified text as a tooltip. The help icon does not show when the label position is "COLLAPSED".
- **required** (Boolean): Determines if a value is required to submit the form. Default: false.
- **disabled** (Boolean): Determines if the user is prevented from interacting with the field. Default: false.
- **validations** (List of Text String): Validation errors to display below the field when the value is not null.
- **height** (Text): Determines the layout height. Valid values: "SHORT", "MEDIUM", "TALL", "AUTO" (default).
- **showWhen** (Boolean): Determines whether the component is displayed on the interface. When set to false, the component is hidden and is not evaluated. Default: true.
- **mapSvg** (Text): SVG Map Source
- **mapAreaParameters** (Text): JSON representation of list of parameters, one for each element in the SVG you want to interact with, in the format:
  {

  <u>id</u>: elementId, /* Content of the id attribute of the SVG element */
  <u>caption</u>: text /* Content of tooltip that will be shown when hovering the element */
  <u>color</u>: elementColor ("ACCENT","STANDARD","POSITIVE","NEGATIVE","SECONDARY" or in format "#hexCode") /* custom color for the SVG element overriding original color */
  <u>selectable</u>: true/false /* Boolean to specify whenever element is clickable or not */
  <u>nodeValues</u>: dictionary/map /* dictionary containing key: value pairs where key correspond on the template key in the format ###key### in referred SVG text node to be replaced with corresponding value. This parameter applies only on text nodes */

  }

- **hoverEffect** (Text): Highlight behavior when mouse hovers on SVG area possible values are "OUTLINE" (default) area will have outline border of same color, "LIGHT" area will be filled with 20% light version of color, "LIGHTER" area will be filled with 50% light version of color, "DARK" area will be filled with 20% dark version of color, "DARKER" area will be filled with 50% dark version of color, "NONE" no highlight behavior
- **enableZoom** (Boolean): Enable Zoom feature in SVG this can be controlled by gesture or mouse wheel
- **enablePan** (Boolean): Enable Pan of SVG Map
- **enableZoomControls** (Boolean): Enable Zoom Control Buttons (enableZoom should be true to display zoom control buttons)
- **zoomControlsIcons** (Boolean): When true display zoom in/out and reset controls as icons instead of buttons

- **zoomControlsPosition** (Text): Position of Zoom Controls , possible values are TOP and BOTTOM which respectively put the control button on above or below SVG
- **zoomControlsAlign** (Text): Alignment of Zoom Control respect the component size , possible values are LEFT, CENTER and RIGHT
- **zoomControlsColor** (Text): Color for Zoom Control Buttons, it can be Appian standard Color (ACCENT,POSITIVE,NEGATIVE,STANDARD) or Hex Color in format #RRGGBB ,if null ACCENT Color is used.
- **maxScale** (Integer): allow to set the maximum zoom level possible values are in the range 2-10, default value is 4
- **svgViewBox** (Text): JSON representation of the svg viewBox attribute parameters in order to override or setup viewBox attribute, the format is :

  ```
  {
      minX: number, /* top left X coordinate of the SVG within the viewBox */
      minY: number  /* top left Y coordinate of the SVG within the viewBox */
      width:number  /* width of svg viewBox */
      height: true/false /* height of svg viewBox */
  }
  ```
-
- **imageStorageConnectedSystem** (Connected System): Reference to Rich Text Upload Connected System to Upload Image (https://community.appian.com/b/appmarket/posts/rich-text-editor-image-upload-connected-system)
- **enableCaptureImage** (Boolean): Flag (true/false) to enable/disable SVG conversion to PNG default is false
- **docId** (List of Save): Document ID of the PNG Version of the Map
- **selectedIndex** (integer): Index of the clicked element in the *mapAreaParameters* structure

Technical Note

To be shown SVG file should start with <svg> tag (no xml/html headers tag)

Every element that needs to be mapped as parameter should be referenced with unique id attribute (e.g., <polygon id="Area1"></polygon>)

<svg> tag should not contain width and height attributes if you want to properly handle standard *height* parameter.

The viewbox attribute of svg tag may need to be adjusted to prevent caption cut as may exceed the area of layout component.

<text> Node template support

Component version 3.0 introduces support to replace the keys denoted with *###* placeholder in SVG <text> node with values passed as parameter in the *nodeValues* structure documented above. This allows for example to insert into the SVG Map KPIs values or names from Appian data for more contextual visualization within the map.

Example: You may have an SVG Map of an office floor with desks layout and each desk element has in the SVG a <text> element which is expected to host the name of assigned employee.

In such case in the SVG file you may change the element in the format *<text id='desk1'>###employee##</text>* and the component will parse the parameter *mapAreaParameters* set to *{id: 'desk1', nodeValues: {employee: local!employees[1]}}* replacing the *>###employee##* content with the *local!employees[1] than* showing that value on the map

SVG to PNG conversion support

In order to use the functionality to export to PNG the SVG Map the Rich Text Upload Connected System to Upload Image (https://community.appian.com/b/appmarket/posts/rich-text-editor-image-upload-connected-system) is required.

Browser/Mobile Support: Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari, Appian for Mobile **Language Support**: English (United States) **Vendor and Support Contacts**: Appian does not provide official support for this component plug-in. Appian (www.appian.com)