Making use of MilStdSymbol

1 Overview

`MilStdSymbol` is the object type returned from `WebRenderer.RenderMultiPointAsMilStdSymbol()`.

1.1 Getting Shapes

Within `MilStdSymbol` there are 3 relevant methods to access data for plotting the symbol to a map.

1.1.1 `MilStdSymbol.getSymbolShapes(): ArrayList<ShapeInfo>`

Get `ArrayList` of `ShapeInfo` containing polylines

1.1.2 `MilStdSymbol.getModifierShapes(): ArrayList<ShapeInfo>`

Get `ArrayList` of `ShapeInfo` containing text and image modifiers

1.1.3 `MilStdSymbol.getTextColor(): Color`

Get `Color` for all text in symbol

1.2 Getting Metadata

These methods are not necessary to plot the symbol but might contain useful information.

1.2.1 `MilStdSymbol.getSymbolID(): String`

20-30 digit code corresponding to a symbol

1.2.2 `MilStdSymbol.getUUID(): String`

User defined from input

1.2.3 `MilStdSymbol.get_WasClipped(): boolean`

True if the rendered symbol fit inside the bounding box

2 Drawing to Map

2.1 Drawing Polylines from `MilStdSymbol.getSymbolShapes()` `ShapeInfo` object

All polylines in a shape have the same properties/style.

2.1.1 `ShapeInfo.getPolylines(): ArrayList<ArrayList<Point2D>>`

For each polyline move to the first point then draw a line to every following point

2.1.2 `ShapeInfo.getStroke().getLineWidth(): float`

Stroke size for all polylines in shape.

2.1.3 `ShapeInfo.getStroke().getEndCap(): int`

Result is `BasicStroke.CAP_BUTT` , `BasicStroke.CAP_ROUND` or `BasicStroke.CAP_SQUARE`

2.1.4 `ShapeInfo.getStroke().getLineJoin(): int`

Result is `BasicStroke.JOIN_MITER` , `BasicStroke.JOIN_ROUND` or `BasicStroke.JOIN_BEVEL`

2.1.5 `ShapeInfo.getStroke().getMiterLimit(): float`

Only relevant if `ShapeInfo.getStroke().getLineJoin()` is `BasicStroke.JOIN_MITER`

2.1.6 `ShapeInfo.getStroke().getDashArray(): float[]`

If attribute `MilStdAttributes.UseDashArray` was set to "true" (can check with `MilStdSymbol.getUseDashArray()`) and `getDashArray()` is not null then apply the dash array to all polylines in shape.

2.1.7 `ShapeInfo.getLineColor(): Color`

Stroke color of all polylines in shape.

2.1.8 `ShapeInfo.getFillColor(): Color`

Fill all polylines in shape with color returned

2.1.9 `ShapeInfo.getShader(): BitmapShader` *Android Only*

Use the shader to fill all polylines in shape. Always returns `null` in Java and TypeScript

2.1.10 `ShapeInfo.getTexturePaint(): TexturePaint` *Java Only*

Use the texture paint to fill all polylines in shape. Always returns `null` in Android and TypeScript.

2.1.11 `ShapeInfo.getPatternFillImage()` : `Bitmap` *(Android)*, `BufferedImage` *(Java)*, or `string` *(TypeScript)*

Use the image to fill all polylines in shape for TypeScript. Optionally can use to get an image object instead of using result of `getShader()` *(Android)* or `getTexturePaint()` *(Java)*.

2.2 Drawing Modifiers from `MilStdSymbol.getModifierShapes()` `ShapeInfo` object

For modifier shapes either `getModifierString()` or `getModifierImage()`will return a nonnull value indicating if the shape is a text or image modifier.

2.2.1 Drawing Text Modifiers

2.2.1.1 `ShapeInfo.getModifierString(): string`

Get text modifier as a string

2.2.1.2 `ShapeInfo.getModifierPosition(): Point2D`

Get text or image position

2.2.1.3 `MilStdSymbol.getTextColor(): Color`

Get the color for the text

2.2.1.4 `RendererSettings.getInstance().getMPLabelFont()` : `Paint` *(Android)* or `Font` *(Java and TypeScript)*

Returned object includes font name, style and size. Alternatively can use the `RendererSettings` methods `getMPModifierFontName()`, `getMPModifierFontType()` and `getMPModifierFontSize()` to get font attributes.

2.2.1.5 `ShapeInfo.getTextJusify(): int`

Result is `ShapeInfo.justify_left` , `ShapeInfo.justify_center` or `ShapeInfo.justify_right`

2.2.1.6 `ShapeInfo.getModifierAngle(): double`

Rotate the text or image by result in degrees

2.2.2 Drawing Image Modifiers

Center justify all image modifiers

2.2.2.1 `ShapeInfo.getModifierImage(): Bitmap` *(Android),* `BufferedImage` *(Java)*, or `string` *(TypeScript)*

Get the image

2.2.2.2 `ShapeInfo.getModifierPosition(): Point2D`

Get text or image position

2.2.2.3 `ShapeInfo.getModifierAngle(): double`

Rotate the text or image by result in degrees