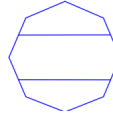


Adding Custom Symbols

It is possible to add custom symbols to be used in unit frames. This feature is limited to the three ports and isn't available in the web service.

1. Decide what your symbol will be and where it will go amongst the existing symbol. This feature will not let you overwrite an existing symbol. For the example, we will add a symbol under Land Unit, Sustainment (symbol set 10, entity code 160000). Sustainment has 56 symbols underneath it so we will add a 57th (165700).



2. Make your custom svg. It should fit within the middle section of the octagon. The octagon svg file is included with the renderer release. Put the shape in a group element with the ID you are going to use. The id is comprised of the Symbol Set Number and the Entity code you are going to assign to this symbol. Below is an example:

```
"<g id=\"10165700\"><text font-family=\"sans-serif\" fill=\"red\" font-size=\"89\" x=\"192\" y=\"428\">MWR</text></g>"
```

3. Next we will add the symbol using calls to the renderer library (Java example)

```
//Get modifiers from an existing symbol in the same symbol set
MSInfo miBase = MSLookup.getInstance().getMSLInfo("10160000",SymbolID.Version_2525E);
//Create new MSInfo object
MSInfo mi = new MSInfo(SymbolID.Version_2525E,"10", "Sustainment","TEST","", "165700",miBase.getModifiers());
//Create new SVGInfo object
SVGInfo si = new SVGInfo("10165700", new Rectangle2D.Double(198.0,365.0,215.0,64.0),"<g id=\"10165700\"><text font-family=\"sans-serif\"
fill=\"red\" font-size=\"89\" x=\"192\" y=\"428\">MWR</text></g>");
//Add symbol to the renderer
MilStdIconRenderer.getInstance().AddCustomSymbol(mi,si);
```

4. Your symbol can now be drawn like any other unit:

```
String svgCustom = MilStdIconRenderer.getInstance().RenderSVG("130310000016570000000000000000",modifiers, attributes)
```

