

# Registering custom symbol renderer provider

The following example shows the custom symbol renderer provider, which provides the *SlotRenderer* for slot symbols, the *PackageRenderer* for package symbols, and the *DependencyRenderer* for a dependency link symbols. The created renderer provider must be registered into the [com.nomagic.magicdraw.uml.symbols.PresentationElementRendererManager](#).

## Step 1. Creating the RendererProvider class

```
/**
 * A custom renderers provider.
 */
class RendererProvider implements PresentationElementRendererProvider
{
    private SlotRenderer slotRenderer;
    private DependencyRenderer dependencyRenderer;
    private PackageRenderer packageRenderer;

    RendererProvider()
    {
        slotRenderer = new SlotRenderer();
        dependencyRenderer= new DependencyRenderer();
        packageRenderer = new PackageRenderer();
    }

    public PresentationElementRenderer getRenderer(PresentationElement
presentationElement)
    {
        if (presentationElement instanceof SlotView ||
presentationElement instanceof SlotListElementView)
        {
            // A slot renderer
            return slotRenderer;
        }

        if (presentationElement instanceof DependencyView)
        {
            // dependency renderer
            return dependencyRenderer;
        }

        if (presentationElement instanceof PackageView)
        {
            // package renderer
            return packageRenderer;
        }

        return null;
    }
};
```

## Step 2. Registering RendererProvider

```
PresentationElementRendererManager.getInstance().addProvider(new
RendererProvider());
```