

Creating an execution engine descriptor

The following example shows how to create an Execution Engine Descriptor

```
public interface ExecutionEngineDescriptor {
    String getEngineName();
    ImageIcon getEngineIcon();
    boolean canExecute(Element element);
    ExecutionEngine createEngine();
    boolean canAnimate(PresentationElement element);
    boolean isAutoDiagramOpened();
    boolean canDebug();
    boolean canUserTriggerEvents();
    boolean isDiagramPerSession();
    boolean isHasIdle();
    boolean isFindEngine(Element element);
    List<? Extends ValidationSuite> getValidationSuite();
}
```

The new execution engine descriptor must implement the ExecutionEngineDescriptor or extend the AbstractExecutionEngineDescriptor abstract Class. You can find details about these Classes in JavaDoc.

```
public class MyExecutionEngineDescriptor extends AbstractExecutionEngineDescriptor {
    @Override

    public boolean canExecute(Element element) {
        return element instanceof Interaction;
    }

    @Override
    public ExecutionEngine createEngine() {
        return new MyExecutionEngine(this);
    }

    @Override
    public String getEngineName() {
        return "My Execution Engine";
    }
}
```

The created execution engine descriptor must then be registered to the simulation manager (see [Registering an execution engine to the Simulation Manager](#)).