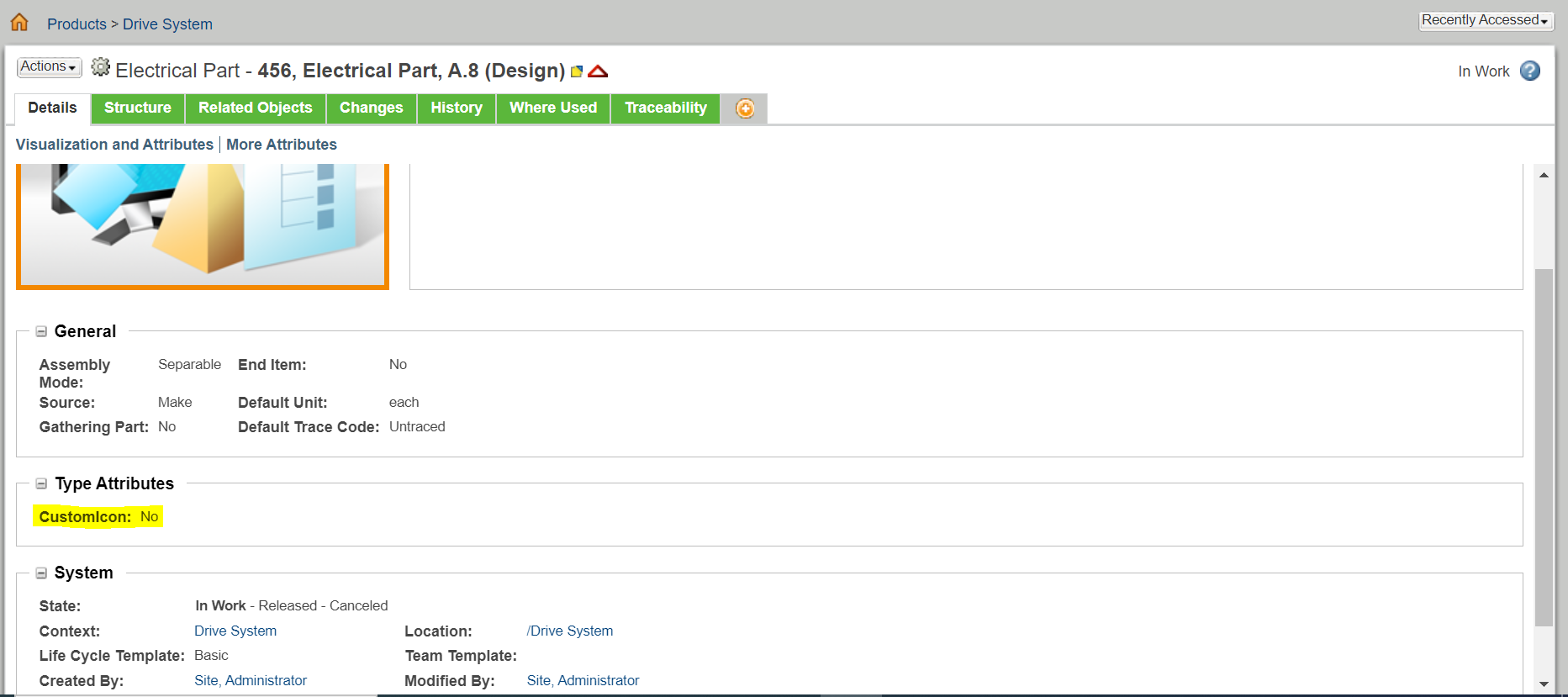
**Status Glyph on Custom Action**

**Use Case:** Displaying the status glyph based on the custom action.

**Step 1: Create custom attribute with datatype Boolean on object:** 

**Step 2:** **Add an entry in custom-actions.xml:**

<action name="customicon" type="object" required="true">

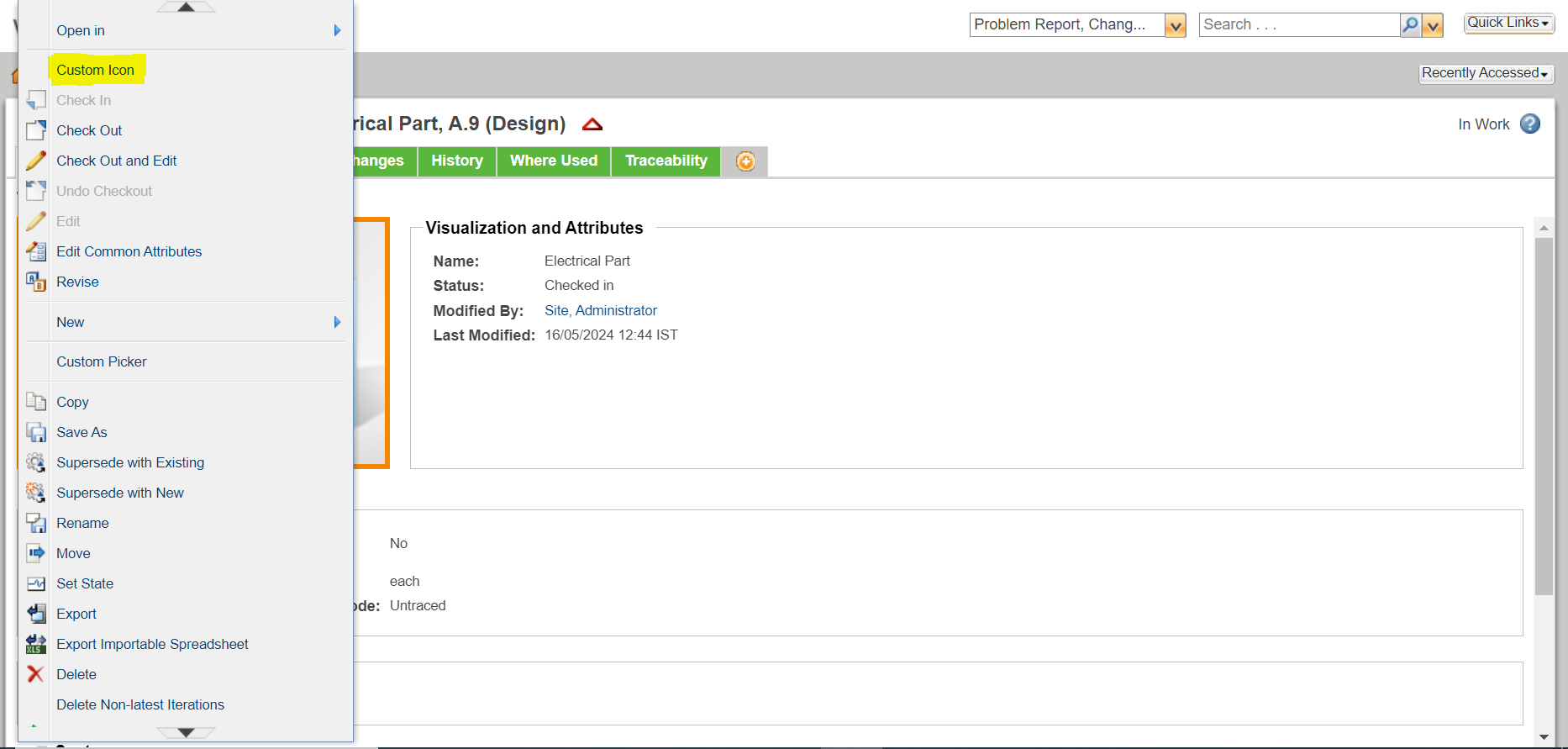
<label>Custom Icon</label>

<command class="ext.plxworx.StatusGlyph" method="execute" onClick="this.disabled=true"/>

</action>

**Custom-actionModel.xml:**

<action name="customicon" type="object"/>



**Step 3:** **Create Data Utility class:**

public class CustomIconDataUtility extends DefaultDataUtility{

public Object getDataValue(String componentId,Object datum,ModelContext modelContext) throws WTException{

Object component=super.getDataValue(componentId,datum,modelContext);

Object customAttrvalue = null;

if(datum instanceof WTPart)

{

WTPart part=(WTPart) datum;

PersistableAdapter obj;

obj = new PersistableAdapter(part, null, Locale.US, null);

obj.load("customicon");

customAttrvalue = obj.get("customicon");

System.out.println("custom value="+customAttrvalue);

if(customAttrvalue.equals(true))

{

String img\_src = "netmarkets\\images\\approbt.gif";

IconComponent iconComponent = new IconComponent(img\_src);

ArrayList arrlst = new ArrayList();

arrlst.add(iconComponent);

return arrlst;

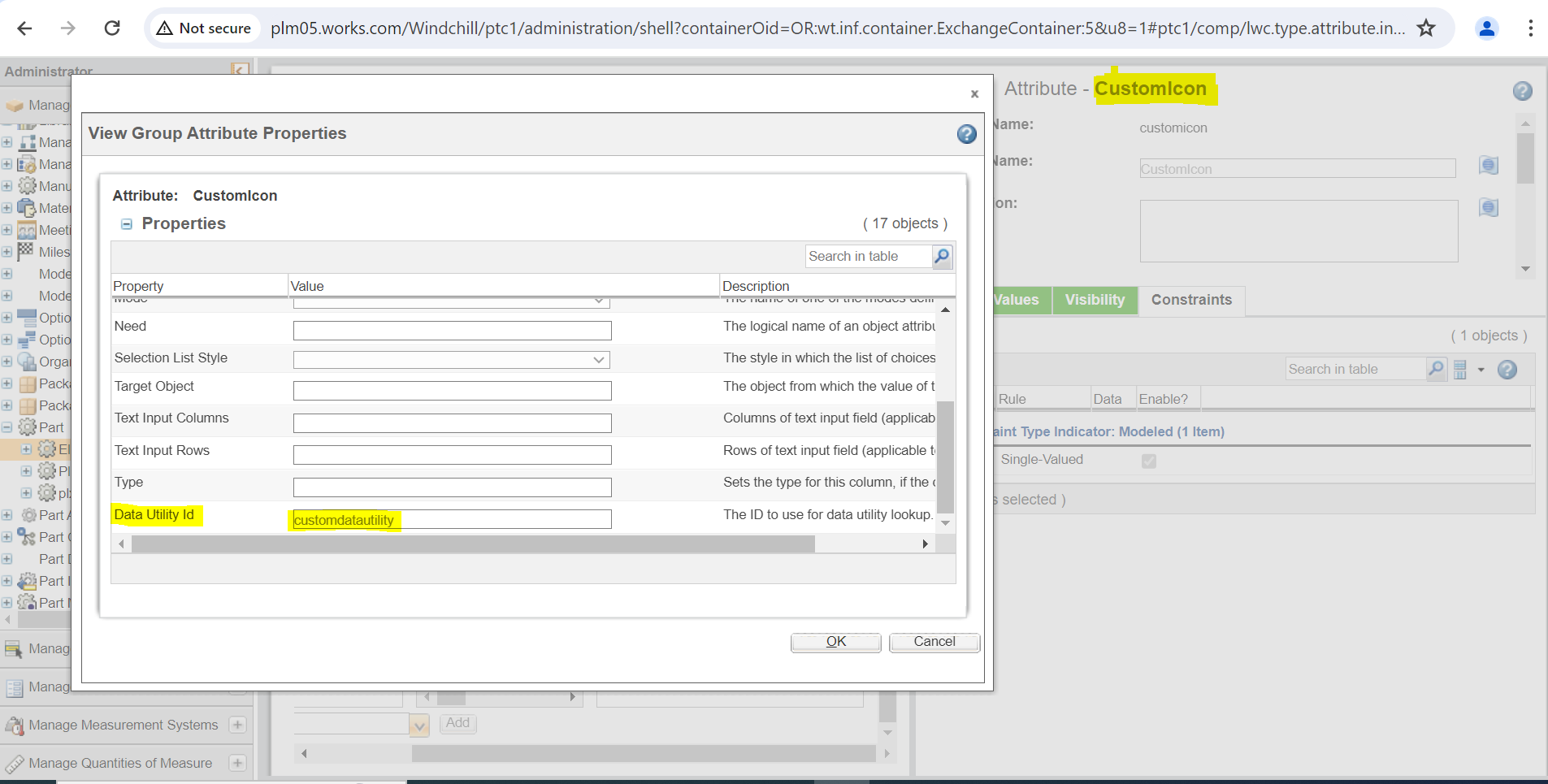
}

return component;

}

}

**Step 4: In Type and Attribute management, add the data utility Id on custom icon.**



**Step 5: Register the service of data Utility class in site.xconf file:**

<Service name="com.ptc.core.components.descriptor.DataUtility" targetFile="codebase/service.properties">

<Option serviceClass="ext.plxworx.CustomIconDataUtility" requestor="java.lang.Object" selector="customdatautility" cardinality="duplicate"/>

</Service>

**Run the command**: xconfmanager –p

**Step 6: Create custom class and extend this OOTB class DefaultInfoComponentBuilder class. Add custom status glyph and set the datautilityId.**

@ComponentBuilder({"infoPage"})

@TypeBased({"com.ptc. ElectricalPart"})

public class ChangeIcon extends DefaultInfoComponentBuilder {

@Override

public InfoConfig buildInfoConfig(ComponentParams var1) throws WTException {

System.out.println("Info page config begin");

InfoComponentConfigFactory var2 = this.getComponentConfigFactory();

InfoConfig var3 = var2.newInfoConfig();

List var4 = var2.getStandardStatusConfigs();

PropertyConfig var5 = var2.newPropertyConfig("netmarkets\\images\\approbt.gif");

var5.setStatusGlyph(true);

var5.setDataUtilityId("customdatautility");

var4.add(var5);

Iterator var6 = var4.iterator();

while (var6.hasNext()) {

ComponentConfig var7 = (ComponentConfig)var6.next();

var3.addComponent(var7);

}

var3.setNavBarName("third\_level\_nav\_part");

var3.setHelpContext("part.view");

var3.setTabSet("partInfoPageTabSet");

return var3;

}

}

**Step 7: Create class for the action:**

StatusGlyph.java:

public class StatusGlyph extends DefaultObjectFormProcessor {

@Override

public FormResult doOperation(NmCommandBean nmCommandBean, List<ObjectBean> list) throws WTException {

WTPart part = (WTPart) nmCommandBean.getPageOid().getRefObject();

System.out.println("parent part:-" + part.getNumber());

findPrtNo(part);

FormResult formResult = super.doOperation(nmCommandBean, list);

//Refresh current page

formResult.setNextAction(FormResultAction.REFRESH\_CURRENT\_PAGE);

System.out.println("page refresh....");

return formResult;

}

private void findPrtNo(WTPart part) throws WTException {

PersistableAdapter persistableAdapter = new PersistableAdapter(part, null, SessionHelper.getLocale(), OperationIdentifier

.newOperationIdentifier(OperationIdentifierConstants.UPDATE));

persistableAdapter.load("customicon");

System.out.println("custom attribute value is loaded.");

Boolean existingValue = (Boolean) persistableAdapter.get("customicon");

// Define flag to determine if the value should be updated

boolean flag = false;

// Define updated value

Boolean updatedValue = null;

// Check the existing value and set the updated value accordingly

if (existingValue != null) {

if (existingValue) {

updatedValue = false;

System.out.println(" If existing value is true, set to false");

flag = true;

} else {

updatedValue = true;

System.out.println("If existing value is false, set to true");

flag = true;

}

} else {

System.out.println("Existing value is null.");

}

// If the flag is true, update the value

if (flag) {

persistableAdapter.set("customicon", updatedValue);

Persistable persistable = persistableAdapter.apply();

PersistenceServerHelper.manager.update(persistable);

AttributeContainer attributeContainer = new IBAValueDBService().updateAttributeContainer((IBAHolder) persistable, persistableAdapter, null, null);

((IBAHolder) persistable).setAttributeContainer(attributeContainer);

System.out.println("Attribute Values updated Successfully");

} else {

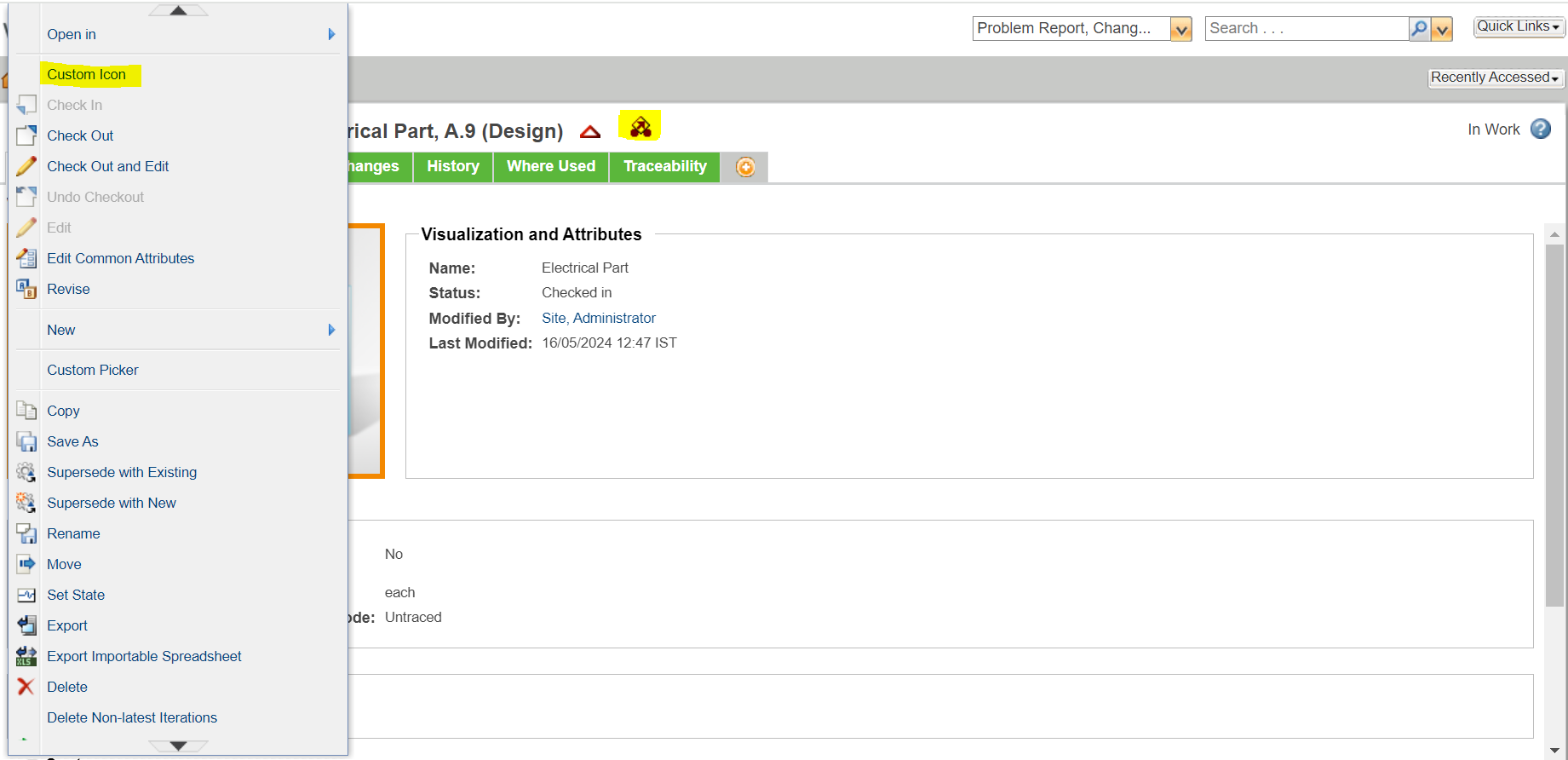
System.out.println("No update needed.");

}

}

}

**Step 8: Output:**

When click on the action then custom status Glyph displayed on the UI. 

Again, clicking on the action custom status glyph will be disabled.

