Scripting Hooks

Introduction

Hooks are midPoint mechanism for injecting a code to or intercepting of normal midPoint flows. E.g. a hook can be used to inset custom notification after an operation is complete, modify the operation request before it is even processed, redirect processing to a workflow or trouble-ticket system or do any other kind of advanced customization magic.

Hooks are typically pieces of Java code that has to be compiled and assembled into midPoint. However there is also a lightweight way to create a hook using a scripting code. Each hook is simply a piece of script code that is executed at the specified moment in the midPoint recompute and execute flow (see Clockwork and Projector). The code can influence the recomputation and execution by manipulating the model context, it can invoke script expression functions, invoke external functions or do almost any other thing.

Scripting Hook Definition

Scripting hooks can be defined in System Configuration Object. The model`Hooks` element is used the define both heavyweight Java hooks and lightweight scripting hooks. The following code block provides an example of Groovy scripting hook.

```
<systemConfiguration oid="00000000-0000-0000-0000-000000000001">
    <name>SystemConfiguration</name>
    <modelHooks>
        <change>
            <hook>
                <name>Example scripting hook</name>
                <state>primary</state>
                <focusType>c:UserType</focusType>
                <script>
                    <code>
                        import com.evolveum.midpoint.prism.delta.*;
                        import com.evolveum.midpoint.xml.ns._public.common.common_3.*;
                        //import com.evolveum.midpoint.model.intest.util.StaticHookRecorder;

                        log.debug('ORG HOOK WAS HERE')

                        UserType user = (UserType)focus;
                        organizations = user.getOrganization();
                        for (orgName in organizations) {
                            org = midpoint.searchObjectByName(OrgType.class, orgName)
                            if (org == null) {
                                // Org does not exist, lets create it
                                org = midpoint.createEmptyObjectWithName(OrgType.class, orgName);
                                topOrgRef = new ObjectReferenceType();
                                topOrgRef.setOid('80808080-8888-6666-0000-100000000001');
                                org.getParentOrgRef().add(topOrgRef);
                                midpoint.addObject(org);
                            }
                            if (!midpoint.isDirectlyAssigned(org)) {
                                // The org is not assigned. Let's assign it.
                                // We need to construct a delta to do this
                                assignment = new AssignmentType();
                                orgTarget = new ObjectReferenceType();
                                orgTarget.setOid(org.getOid());
                                orgTarget.setType(OrgType.COMPLEX_TYPE);
                                assignment.setTargetRef(orgTarget);
                                assignmentDelta = ContainerDelta.createModificationAdd(UserType.F_ASSIGNMENT,
UserType.class, prismContext, assignment);
                                modelContext.getFocusContext().swallowToPrimaryDelta(assignmentDelta);
                            }
                        }
                    </code>
                </script>
            </hook>
        </change>
    </modelHooks>
    ....
</systemConfiguration>
```

Copyright © 2011-2020 Evolveum. All rights reserved.

Example 1

Following scripting hook removes all assignments from disabled users. Please note usage of modelContext.rot()

```
<hook>
    <name>Remove assignments from disabled users</name>
    <state>secondary</state>
    <focusType>c:UserType</focusType>
    <script>
       <code>
            import com.evolveum.midpoint.prism.delta.*;
            import com.evolveum.midpoint.xml.ns._public.common.common_3.*;

            UserType user = (UserType) focus;
            ActivationStatusType administrativeStatus = user.getActivation().getAdministrativeStatus();
            if (administrativeStatus == ActivationStatusType.DISABLED) {
                                changed = false;
                for (AssignmentType assign : user.getAssignment()) {
                    assignmentDelta = ContainerDelta.createModificationDelete(UserType.F_ASSIGNMENT, UserType.
class, prismContext, assign.clone());
                    log.debug('Removing assignment ' + assignmentDelta + ' from disabled user ' + user.
getName());
                    modelContext.getFocusContext().swallowToSecondaryDelta(assignmentDelta);
                                        changed = true;
                }
                                if (changed) {
                    modelContext.rot(); // this makes Projector to recompute the model context
                }
            }
        </code>
    </script>
</hook>
```

Copyright © 2011-2020 Evolveum. All rights reserved.