**Life Cycle Event-Joiner-AD**

Lifecycle Event

With LCM enabled, Lifecycle Events can be configured in IdentityIQ to represent activities that happen in the normal course of a person’s employment at a company: events like joining the company, changing departments/managers, leaving the company. These events are also referred to by the shorthand terms: Joiner, Mover, Leaver.

When LCM is enabled, IdentityIQ contains four pre-defined Lifecycle Events.

• Joiner

• Leaver

• Manager Transfer

• Reinstate

All of these are disabled by default and must be enabled before they will be triggered. Lifecycle Events are triggered by specific changes to an Identity – creation, manager transfer, attribute change, or more complex changes detected by an IdentityTrigger Rule. They invoke Business Processes, or Workflows, which may contain provisioning actions, if desired.

NOTE: The terms Business Process and Workflow are synonymous. The IdentityIQ user interface refers to them as Business Processes -- the term most often used by business managers. Behind the scenes, in the IdentityIQ object model and XML, they are called Workflows; they control the flow of data through the required processing.

The pre-defined Lifecycle Events function, by default, as shown in the table below.

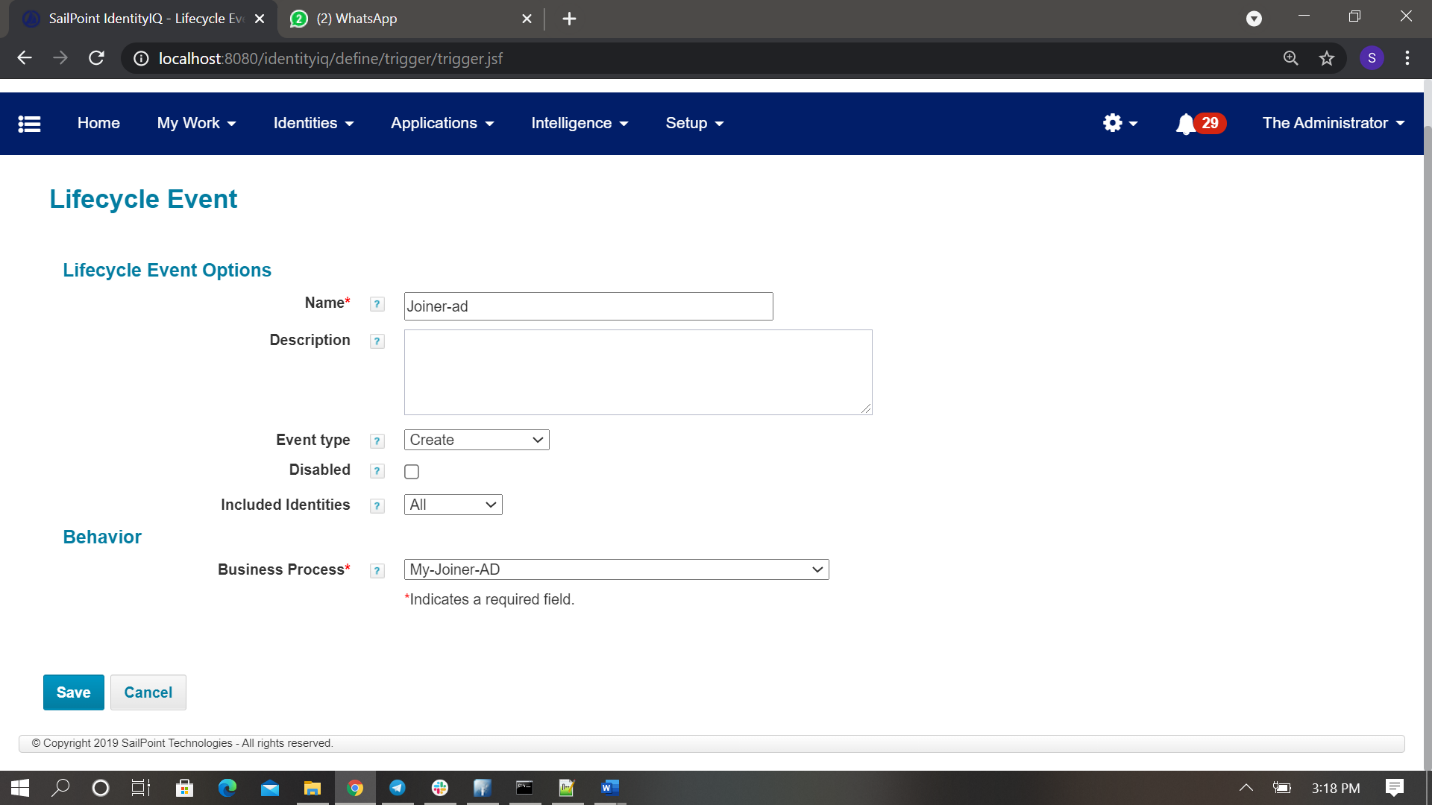
|  |  |  |
| --- | --- | --- |
| Lifecycle Event | Trigger | Business Process |
| Joiner | Identity Creation | Lifecycle Event - Joiner |
| Leaver | Attribute Change: “Inactive” attribute change from false to true | Lifecycle Event – Leaver |
| Manager Transfer | Manager Change | Lifecycle Event – Manager Transfer |
| Reinstate | Attribute Change: “Inactive” attribute change from true to false | Lifecycle Event – Reinstate |

Here we can take example of Joiner-AD.

Add an Employee in Hr-Application.csv.

Make sure that the Active Directory Server is opened.

On home page **Setup**-->**Life Cycle Events-->Add New Life Cycle Event.**



Name : Joiner-AD

Event type : Create

Business Process : MyJoiner-Ad-3.

Here the Business process is nothing but workflow. Workflow related to joiner-AD should be selected.

**Save.**

After that we need to run HR Aggregation task, Refresh Identity Cube task.

In **Refresh Identity Cube** task we need to select the option **Process events.**

**We can run the Refresh Identity Cube** task for particular person also by Optional filter string to constrain the identities refreshed name==”AD User”.

If needed perform maintenance task should be run (The identity is in provision.)

We can see the identity in Identity cube.

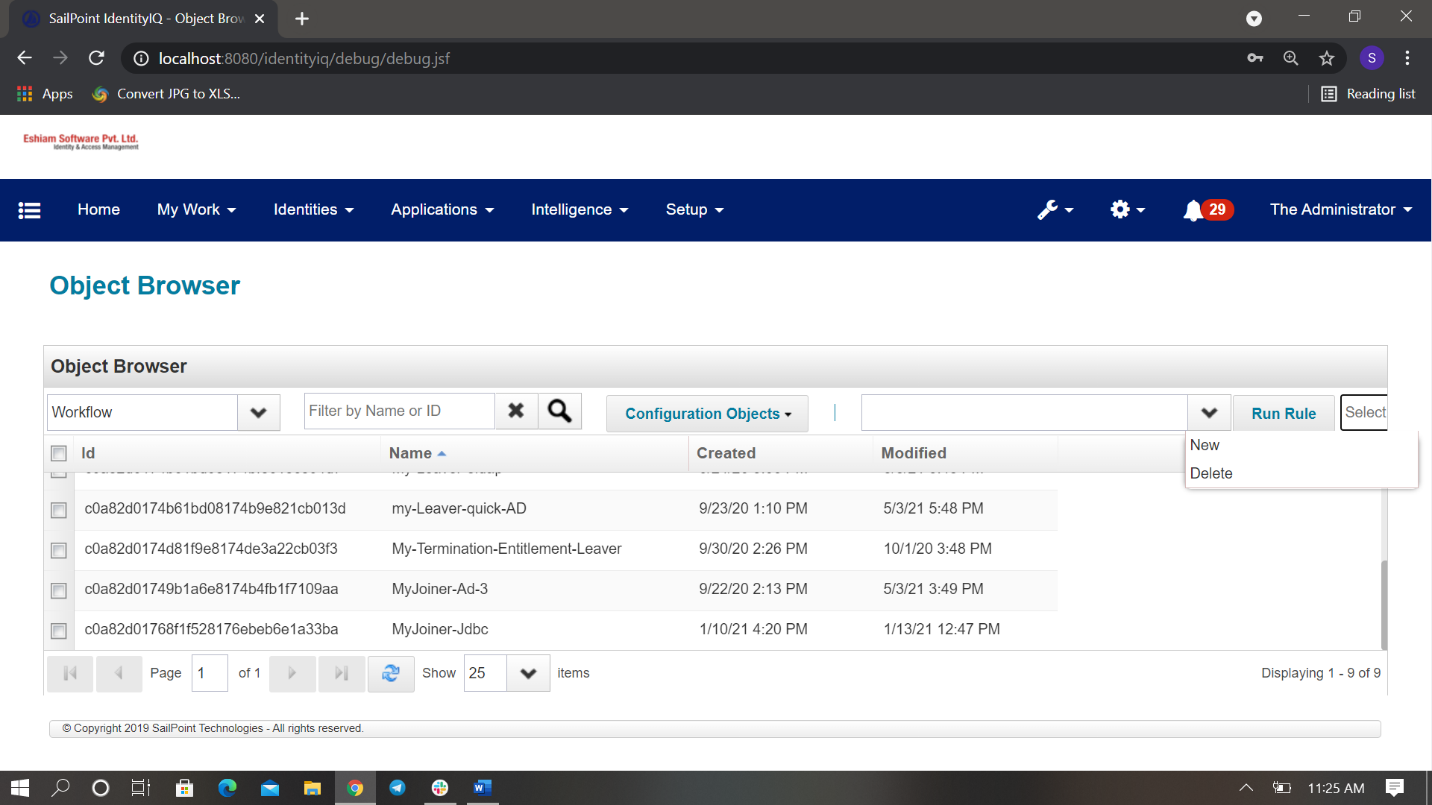
**NOTE: Here while we are selecting the Business Process the workflow needs to be define.**

**Go to Debug page-->workflow-->new-->Paste the below code -->save.**

acctReqAD.setApplication("AD");

Here we need choose the Application name.

Business Process name is workflow name.





<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE Workflow PUBLIC "sailpoint.dtd" "sailpoint.dtd">

<Workflow created="" explicitTransitions="true" id=" " libraries="Identity" modified="" name="MyJoiner-Ad-3" type="IdentityLifecycle">

<Variable input="true" name="trigger">

<Description>The IdentityTrigger</Description>

</Variable>

<Variable initializer="joiner" name="flow">

<Description>The flow</Description>

</Variable>

<Variable initializer="joiner" input="true" name="trace"/>

<Variable input="true" name="event" transient="true">

<Description>

The IdentityChangeEvent. It can be used to build

the provisioning plan, but does not need to be

persisted with the case, so marked as transient.

</Description>

</Variable>

<Variable input="true" name="identityName">

<Description>The name of the identity.</Description>

</Variable>

<Description>Process a new employee.</Description>

<Step icon="Start" name="Start" posX="28" posY="10">

<Transition to="Process user"/>

</Step>

<Step icon="Message" name="Process user" posX="148" posY="10" resultVariable="adPlan">

<Description>Process the user that joined.</Description>

<Script>

<Source>

import sailpoint.object.\*;

import sailpoint.api.\*;

System.out.println("New user was entered the system: " + event.getObject().getDisplayableName());

Identity identityObj = context.getObjectByName(Identity.class,identityName);

ProvisioningPlan plan = new ProvisioningPlan();

plan.setIdentity(identityObj);

plan.setNativeIdentity(identityObj.getName());

AccountRequest acctReqAD = new AccountRequest();

acctReqAD.setApplication("AD");

acctReqAD.setOp(ProvisioningPlan.ObjectOperation.Create);

String nativeIdentity = "CN="+identityObj.getName()+","+"OU=People,OU=Demo,DC=Eshiam,DC=com";

acctReqAD.add(new AttributeRequest("samAccountName", ProvisioningPlan.Operation.Add,identityObj.getName()));

acctReqAD.add(new AttributeRequest("distinguishedName", ProvisioningPlan.Operation.Add,nativeIdentity));

acctReqAD.add(new AttributeRequest("employeeID", ProvisioningPlan.Operation.Add,identityObj.getAttribute("employeeID")));

acctReqAD.add(new AttributeRequest("givenName", ProvisioningPlan.Operation.Add,identityObj.getFirstname()));

acctReqAD.add(new AttributeRequest("sn", ProvisioningPlan.Operation.Add,identityObj.getLastname()));

acctReqAD.add(new AttributeRequest("memberOf", ProvisioningPlan.Operation.Add,"CN=AllUsers,OU=Group,OU=Demo,DC=Eshiam,DC=com"));

plan.add(acctReqAD);

return plan;

</Source>

</Script>

<Transition to="provisionUser"/>

</Step>

<Step icon="Task" name="provisionUser" posX="183" posY="12">

<Arg name="approvalSet" value="ref:approvalSet"/>

<Arg name="flow" value="ref:flow"/>

<Arg name="identityName" value="ref:identityName"/>

<Arg name="launcher" value="ref:launcher"/>

<Arg name="plan" value="ref:adPlan"/>

<Arg name="approvalScheme" value="none"/>

<Arg name="foregroundProvisioning" value="true"/>

<Return name="project" to="project"/>

<WorkflowRef>

<Reference class="sailpoint.object.Workflow" id="" name="LCM Provisioning"/>

</WorkflowRef>

<Transition to="Stop"/>

</Step>

<Step icon="Stop" name="Stop" posX="268" posY="10"/>

</Workflow>

**Thankyou**