**Notes:**

**1.check test connection**

try {

Application app = (Application) context.getObject(Application.class, "Global Pass");

if (app != null) {

Connector connector = ConnectorFactory.getConnector(app, null);

connector.testConfiguration();

}

}catch (Exception e) { }

**2. Single Account Aggeregation:**

Application application = context.getObjectByName(Application.class, "AD app");

IdentityService service = new IdentityService(context);

ResourceObject resourceObject = new ResourceObject();

resourceObject.setIdentity("CN="+identityName+",CN=Users,DC=seri,DC=sailpointdemo,DC=com");

resourceObject.setObjectType("account");

resourceObject.setDisplayName(identityName);

System.out.println(resourceObject.toXml());

ADLDAPConnector appConnector = new ADLDAPConnector(application);

Attributes argMap = new Attributes();

argMap.put("applications", application.getName());

argMap.put("checkDeleted",true);

Aggregator agg = new Aggregator(context, argMap);

agg.aggregate(application, resourceObject);

**3.Terminator Object:**

Filter f = Filter.ignoreCase(Filter.eq("name", "HLTN Sunset WF - "+identityName));

QueryOptions qo = new QueryOptions();

qo.addFilter(f);

List requests = context.getObjects(Request.class, qo);

if(requests != null && !requests.isEmpty()){

Terminator terminator = new Terminator(context);

for(Request request : requests){

lcmlogger.trace("removing sunset task for "+identityName);

terminator.deleteObject(request);

}

context.commitTransaction();

}

**4.Dynamic Scope to add Identity:**

**🡪**add in the Dynamic Scope object to add Identity

<Inclusions>

<Reference class="sailpoint.object.Identity" id="0a542f7b6c901bc2816c99e8c657194d" name="agesesse"/>

</Inclusions>

**6.Create LDAP OU:**

**\*create LDAP OU’s and groups if not Exist**

import java.util.Hashtable;import javax.naming.Context;import javax.naming.NamingEnumeration;import javax.naming.NamingException;import javax.naming.directory.Attribute;import javax.naming.directory.Attributes;import javax.naming.directory.BasicAttribute;import javax.naming.directory.BasicAttributes;import javax.naming.directory.DirContext;import javax.naming.directory.InitialDirContext;import javax.naming.directory.SearchControls;import javax.naming.directory.SearchResult; String[] attrIDs = { "sn", "cn" }; SearchControls ctls = new SearchControls(); ctls.setReturningAttributes(attrIDs); String ouName = "NewOU"; String domainName = "dc=maxcrc,dc=com"; String filter = "(ou=" + ouName + ")"; //return filter; Hashtable env = new Hashtable (); env.put(Context.INITIAL\_CONTEXT\_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory"); // env.put(Context.PROVIDER\_URL, "yourURL"); env.put(Context.PROVIDER\_URL, "ldap://localhost:389"); env.put(Context.SECURITY\_AUTHENTICATION, "simple"); env.put(Context.SECURITY\_PRINCIPAL, "cn=Manager,dc=maxcrc,dc=com"); env.put(Context.SECURITY\_CREDENTIALS, "secret"); DirContext ctx = new InitialDirContext(env); NamingEnumeration e = ctx.search(domainName, filter, ctls); if (e.hasMore()) { System.out.println("inside ifin ou"); while (e.hasMore()) { SearchResult entry = (SearchResult) e.next(); System.out.println(entry.getName()); } } else { System.out.println("inside else"); Attributes attrs = new BasicAttributes(true); // case-ignore Attribute objclass = new BasicAttribute("objectclass"); // objclass.add("top"); objclass.add("organizationalUnit"); // attrs.put(objclass); attrs.put("ou", "smith"); attrs.put("objectClass", "organizationalUnit"); DirContext createSubcontext = ctx.createSubcontext("ou=" + ouName + "," + domainName, attrs); } String groupName = "Test1"; String groupOU = "ou=" + ouName + ","; String groupFilter = "(cn=" + groupName + ")"; e = ctx.search(groupOU + domainName, groupFilter, ctls); if (e.hasMore()) { System.out.println("inside if in group"); while (e.hasMore()) { SearchResult entry = (SearchResult) e.next(); System.out.println(entry.getName()); } } else { System.out.println("inside else in group"); Attributes attrs = new BasicAttributes(true); // case-ignore //Attribute objclass = new BasicAttribute("objectclass"); // objclass.add("top"); // objclass.add("organizationalUnit"); // attrs.put(objclass); attrs.put("ou", ouName); attrs.put("objectClass", "groupOfUniqueNames"); attrs.put("uniqueMember", "cn=Robert Smith,ou=People,dc=maxcrc,dc=com"); DirContext createSubcontext = ctx.createSubcontext("cn=" + groupName + "," + groupOU + domainName, attrs); }return "Ok";

**7.Email with attachment:**

byte[] csvfileData = buffer.toString().getBytes();

  EmailTemplate emailTemplate = context.getObject(EmailTemplate.class,template);

  EmailFileAttachment attachmentCSV = new EmailFileAttachment(fullReport + ".csv", EmailFileAttachment.MimeType.MIME\_CSV, csvfileData);\\\

EmailOptions emailOptions = new EmailOptions();

  emailOptions.addAttachment(attachmentCSV);

  emailOptions.setVariable("resultMatrices",resultMatrices);

  emailOptions.setTo("[asheesh1.kumar@xx.com](mailto:asheesh1.kumar@xx.com)");

  mylogger.trace("\*\*\*\*\*\*\*\*\*\* Sending email\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ::");

  context.sendEmailNotification(emailTemplate, emailOptions);

**7.Email Notification:**

Map map = new HashMap();EmailTemplate template = context.getObjectByName(EmailTemplate.class, "notification template");

EmailOptions options = new EmailOptions(); options.setTo("Abc"); options.setVariables(map); context.sendEmailNotification(template, options);

<https://community.sailpoint.com/t5/IdentityIQ-Forum/How-to-pass-variable-value-into-emailtemplate-from-report/m-p/149536#M124160>

**8.Launch Workflow:**

Workflow wf = context.getObject(Workflow.class,"Native Change Detection");

WorkflowLaunch wflaunch = new WorkflowLaunch();

wflaunch.setWorkflowName(wf.getName());

//Create Workflower and launch workflow from WorkflowLaunch

Workflower workflower = new Workflower(context);

WorkflowLaunch launch = workflower.launch(wflaunch);

**Schedule Workflow:**

String workflowName = "Native Change Detection";

String caseName = "Run" + workflowName + " for: " + identityName;

Workflow eventWorkflow = context.getObject(Workflow.class, workflowName);

Identity id = context.getObjectByName(Identity.class, "spadmin");

long launchTime = System.currentTimeMillis() + 300000;

caseName = caseName + "(" + launchTime + ")";

Attributes reqArgs = new Attributes();

reqArgs.put(StandardWorkflowHandler.ARG\_REQUEST\_DEFINITION,

sailpoint.request.WorkflowRequestExecutor.DEFINITION\_NAME);

reqArgs.put(sailpoint.workflow.StandardWorkflowHandler.ARG\_WORKFLOW, workflowName);

reqArgs.put(sailpoint.workflow.StandardWorkflowHandler.ARG\_REQUEST\_NAME, caseName);

reqArgs.put("requestName", caseName);

Attributes wfArgs = new Attributes();

wfArgs.put("identityName", identityName);

wfArgs.put("workflow", eventWorkflow.getId());

reqArgs.putAll(wfArgs);

Request req = new Request();

RequestDefinition reqdef = context.getObject(RequestDefinition.class, "Workflow Request");

req.setDefinition(reqdef);

req.setEventDate(new Date(launchTime));

req.setOwner(id);

req.setName(caseName);

req.setAttributes(reqdef, reqArgs);

System.out.println("After request object");

RequestManager.addRequest(context, req);

Schedule Workflow in Step:

<Step action="call:scheduleWorkflowEvent" name="Domain Movement" posX="311">  
    <Arg name="workflow" value="Name of the child workflow"/>  
    <Arg name="requestName" value="A name that would like to see in the task result page"/>  
    <Arg name="scheduleDelaySeconds" value="3 days in seconds"/>  
    <Arg name="*domainMoveType*" value="ref:varName"/>  
    <Transition to="Next step"/>  
  </Step>