**API to find Bundle match criteria : (which identity match criteria)**

Matchmaker matchMaker = new Matchmaker(context);

Bundle bRole = context.getObjectByName(Bundle.class, "Birthright Roles");

IdentitySelector idSelector = bRole.getSelector();

boolean isMatch = matchMaker.isMatch(idSelector,identity);

**Create Bundle through Rule:(entitlement,IT role,Business role)**

public static void createBundle(String managedattribute, String workGroup,String company,String description) throws GeneralException {

log.error("inside createBundle");

List managedattributeList = new ArrayList();

managedattributeList.add(managedattribute);

Attributes roleAttributes = new Attributes();

Map roleMap = new HashMap();

roleMap.put("en\_US", "This Role is tied to company/partner name:"+company);

roleAttributes.put("accountSelectorRules","");

roleAttributes.put("allowDuplicateAccounts","false");

roleAttributes.put("allowMultipleAssignments","false");

roleAttributes.put("mergeTemplates","false");

roleAttributes.put("sysDescriptions",roleMap);

Attributes maAttributes = new Attributes();

Map maMap = new HashMap();

maMap.put("en\_US", "This Entitlement is tied to company/partner name:"+description);

maAttributes.put("sysDescriptions",maMap);

Identity owner = context.getObject(Identity.class, workGroup);

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

if(owner != null &amp;&amp; application != null) {

Filter filter = Filter.eq("value", managedattribute);

QueryOptions query = new QueryOptions();

query.addFilter(filter);

List &lt;ManagedAttribute> objects = context.getObjects(ManagedAttribute.class,query);

if(objects == null || objects.isEmpty()) {

log.error("Entitlement not Exists so creating Entitlement,IT Role, Business Role");

ManagedAttribute ma = new ManagedAttribute();

ma.setValue(managedattribute);

ma.setApplication(application);

ma.setAttribute("memberOf");

ma.setRequestable(true);

ma.setType("Entitlement");

ma.setAttributes(maAttributes);

ma.setOwner(owner);

context.saveObject(ma);

context.commitTransaction();

Bundle it\_bundle = new Bundle();

it\_bundle.setName(managedattribute+"-ITRole");

it\_bundle.setType("it");

it\_bundle.setAttributes(roleAttributes);

it\_bundle.setOwner(owner);

Profile profile = new Profile();

profile.setApplication(application);

LeafFilter leafFilter = new LeafFilter();

leafFilter.setOperation(LogicalOperation.CONTAINS\_ALL);

leafFilter.setProperty("memberOf");

leafFilter.setValue(managedattributeList);

List &lt;Filter> leafFilterList = new ArrayList();

leafFilterList.add(leafFilter);

profile.setConstraints(leafFilterList);

List profilesList = new ArrayList();

profilesList.add(profile);

it\_bundle.setProfiles(profilesList);

context.saveObject(it\_bundle);

context.commitTransaction();

List &lt;Bundle> itBundleList = new ArrayList();

itBundleList.add(it\_bundle);

Bundle bus\_bundle = new Bundle();

bus\_bundle.setName(managedattribute+"-BusinessRole");

bus\_bundle.setType("business");

bus\_bundle.setAttributes(roleAttributes);

bus\_bundle.setOwner(owner);

bus\_bundle.setRequirements(itBundleList);

context.saveObject(bus\_bundle);

context.commitTransaction();

}

else{

log.error("Entitlement Already Exists");

}

}

}