Connectors Examples

- Attribute mappings examples
- Outgoing triggers examples
- Incoming triggers examples
- Triggers: Script Tips

Attribute mappings examples

Attributes

When you are configuring an agent, depending on the connector type, it will be able to define some attributes. The attributes depend on the object that you are configuring, and the objects depend on the connector type.

Get the value of an attribute (a user attribute in that case)

```
sAMAccountName <= userName
sAMAccountName => userName
sAMAccountName <=> userName
```

Get the value of a custom attribute

```
company <= attributes{"company"}
company => attributes{"company"}
company <=> attributes{"company"}
```

Get the value of an account metadata attribute

```
office <= accountAttributes{"office"}
office => accountAttributes{"office"}
office <=> accountAttributes{"office"}
```

Define a constant on the target system

```
nameConst <= "valueConst"
```

Assign the result of a script to the soffid attribute

```
return <SCRIPT> => attribute
```

> Be in mind, it is allowed to use bean Shell expression only in the source when the mapping is one-way.

Assign the result of a script to the target attribute

```
attribute <= return <SCRIPT>
```

> Be in mind, it is allowed to use bean Shell expression only in the source when the mapping is one-way.

Be in mind

> If the attribute name in the final system is hyphenated, you have to use this expression: **THIS{"ATTRIBUTENAME"}**, for example THIS{"my-name"}

Outgoing triggers examples

Attribute mapping triggers

When you are configuring an agent and defining the attribute mappings of connectors, depending on the connector type, it will be able to define BeanShell scripts that will be triggered when data is loaded into the target system, **outgoing triggers**.

Triggers can be used to validate or perform a specific action just before performing an operation or just after performing an operation into target objects.

The trigger result will be a boolean value, true to continue or false to stop.

Use case examples

Example 1

Update or insert a user only when the user is internal (PreInsert User or PreUpdate User)

```
name = source{"userName"};
user = serviceLocator.getUserService().findUserByUserName(name);
if (user != null) {
  if (user.userType.equals("I")) {
      return true;
  }
}
return false;
```

Example 2

Update or insert a user only when the company is Soffid. Be in mind that company attribute is a custom attribute.

```
name = source{"userName"};
company = source{"attributes"}{"company"};
user = serviceLocator.getUserService().findUserByUserName(name);
```

```
if (user != null) {
  if (company != null && company.toUpperCase().equals("SOFFID")) {
   return true;
  }
}

return false;
```

Example 3

Recover a response and process it.

```
if (response != null) {
  for (o : response.getObjects()) {
   if (o != null && o{"result"} != null) {
    //TO-DO
    }
  }
}
return true;
```

Example 4

Send a HTML mail with the response info.

```
.....
to = newObject{"identity"};
subject = "LinOTP QR";
body = o{"result"};
serviceLocator.getMailService().sendHtmlMailToActors(new String[] {to}, subject, body);
.......
```

Example 5

Update a user with the response info

```
.......
body = o{"result"};
ac = newObject{"user"};
data = new com.soffid.iam.api.UserData();
```

```
data.setAccountName(ac);
data.setSystemName("LinOTP");
data.setAttribute("token");
data.setBlobDataValue(body.getBytes());
serviceLocator.getAccountService().updateAccountAttribute(data);
......
```

Example 6

Call an API and process the response.

```
.......
userN = accontList.get(0).name;
result = dispatcherService.invoke(
"GET",
 "https://" + DOMINIO + "crmRestApi/resources/....?onlyData=true&q=Username=" + userN,
 null);
if (result != null && !result.isEmpty()) {
 //TO-DO
}
.......
```

Example 7

Pre Update: assign a value to an attribute depending on the value of a group attribute. The user can belong to many groups.

```
userName = source{"userName"};
attributes = serviceLocator.getUserService().findUserAttributes(userName);
dateChange = attributes.get("dateChange");

if (dateChange == null || dateChange == void ) {
return true;
}

userGroupList = serviceLocator.getGroupService().findUsersGroupByUserName(userName);
contFalse = 0;
contNull  = 0;
isFound = false;
for (ug : userGroupList) {
 group = serviceLocator.getGroupService().findGroupByGroupName(ug.group);
```

```
  att = group.attributes.get("indicator");
  if (att != void) {
     if (att == null) contNull++;
     else if (att.equals("0")) contFalse++;
     else if (att.equals("1")) {
       isFound  = true;
       break;
     }
  }
}

if (isFound) {
  newObject{"id-indicator"} = "1";
} else {
  if (contFalse > 0) {
    newObject{"id-indicator"} = "0";
  } else if (contNull > 0) {
    newObject{"id-indicator"} =  null;
  }
}

return true;
```

Incoming triggers examples

Load triggers

When you are configuring an agent, depending on the connector type, it will be able to define BeanShell scripts that will be triggered when data is loaded into Soffid, **incoming triggers**.

Triggers can be used to validate or perform a specific action just before performing an operation or just after performing an operation into Soffid objects.

The trigger result will be a boolean value, true to continue or false to stop.

Use case examples

Example 1

A user can have more than one account on a target system. We want to reconcile only those accounts which have the same name on Soffid and on target system.

```
name = newObject{"accountName"};
uaList = serviceLocator.getAccountService().findUsersAccounts(name,"agentName");

for(userAccount: uaList) {
  if (userAccount.name.equals(name)) {
    return true;
  } else  {
    return false;
  }
}

return false;
```

Example 2

Update only users who belong to Soffid company.

```
name = newObject{"accountName"};
user = serviceLocator.getUserService().findUserByUserName(name);

if (user != null) {
  attributes = serviceLocator.getUserService().findUserAttributes(name);
  company = attributes.get("company");
  if (company != null && company.equals("Soffid")) {
  return true;
}
return false;
```

Example 3

Discard to create some accounts (PreInsert account).

```
log.info("********** Pre Insert Account");
cuentas = new java.util.HashMap();
cuentas.put("admin",null);
account = newObject{"accountName"};

if (cuentas.containsKey(account)) {
  log.info("TRIGGER ACCOUNT PREINSERT - Discarded to create the account " + account);
  return false;
}

log.info("TRIGGER ACCOUNT PREINSERT - Correct account " + account);
return true;
```

Example 4

If does not exist a mail domain, it wil be created.

```
mailDomain = newObject{"mailDomain"};
if (mailDomain != void && mailDomain != null) {
  existMD = serviceLocator.getMailListsService().findMailDomainByName(mailDomain);
  if (existMD == null) {
    newMailDomain = new com.soffid.iam.api.MailDomain();
    newMailDomain.setName(mailDomain);
    newMailDomain.setDescription(mailDomain);
```

```
    serviceLocator.getMailListsService().create(newMailDomain);
  }
}
return true;
```

Example 5

Avoid deleting users (PreDelete user).

```
return false;
```

Example 6

If throw the exception, the return will be is false. True in other cases.

```
if (....)
  throw new Exception("Exception message....");
return true;
```

Example 7

The new group on the target system will have the same id that the old group.

```
newObject{"idGroup"} = oldObject{"idGroup"};
.....
```

Example 8

Get the attribute company option 1:

```
company = source{"attributes"}{"company"};
```

Get the attribute company option 2

```
userName = source{"userName"};
attributes = serviceLocator.getUserService().findUserAttributes(userName);
company = attributes.get("company");
```

Example 9

Update the company attribute:

```
userName = newObject{"userName"};
// Check if the user exists
user = serviceLocator.getUserService().findUserByUserName(userName);
if (user == null) {
return false;
}
log.info("***************** USER Object: " + user);

attributes = serviceLocator.getUserService().findUserAttributes(userName);
if (attributes == null) {
attributes = new HashMap();
}

attributes.put("company", "<COMPANY_NAME>");
serviceLocator.getUserService().updateUserAttributes(userName, attributes);

return true;
```

Example 10

Check user type

```
userName = newObject{"userName"};
user = serviceLocator.getUserService().findUserByUserName(userName);

if (user.userType.equals("I")){
  .....
  //TODO
} else {
    .....
  //TODO
}
return true;
```

Triggers: Script Tips

Triggers: Script Tips

Here we will show you some tips about how to use scripts.

> For more information you can visit the official documentation of Soffid

Write into a sync-server log

```
System.out.println("what you want.....");
```

Recover data from a Soffid object when synchronizing

> * ***source*** *only can be used in outgoing triggers. That object will be Soffid format.*

> * ***newObject*** *and* ***oldObject*** *can be used in outgoing and incoming triggers. Those objects will be target system format in outgoing triggers and will be Soffid format in incoming triggers.*

Recover data from Soffid object to synchronize

Recover user name

```
name = source{"userName"};
```

Recover a custom attribute

Recover company (company could be a user custom attribute defined on metadata page)

```
comp = source{"attributes"}{"company"};
```

Recover the attribute value that will be sent

```
no = newObject{"userName"};
gn = newObject{"givenName"};
....
```

Recover the attribute value from the select

That info comes from the target system. In a synchronization process, the first thing that Soffid does, is to query if the account exists on the target system.

```
no = oldObject{"userName"};
gn = oldObject{"givenName"};
....
```

Use existing services

```
serviceLocator.getUserService().....
serviceLocator.getAccountService().....
serviceLocator.......
```

Create a new Soffid object

```
mailDomain = new com.soffid.iam.api.MailDomain();
newUser = new com.soffid.iam.api.User();
newRol = new com.soffid.iam.api.Role();
.............
```

Loop an object list

```
for(role : roleList){
  //TO-DO
  out.print(" Role: " + role.roleName + "\n");
}
```

Recover response

```
if (response!=null) {
  for (o : response.getObjects()) {
    if (o!=null && o{"result"}!=null) {
      //TO-DO
    }
```

```
  }
}
```

Send a text email

```
serviceLocator.getMailService().sendTextMail("mail@soffid.com", "Subject", "Mail message");
```

Send a HTML mail

```
serviceLocator.getMailService().sendHtmlMailToActors(<to>, <subject>, <html-body>);
```