Agents

- Managed systems
  - Description
  - Addons installation
  - Configuration
    - Basics
      - Generic parameters
      - Custom attributes
      - Synchronization buttons
    - Attribute Mapping
      - Properties
      - Attributes
        - System attributes
        - Directions
        - Soffid attributes
    - Load triggers
    - Account metadata
  - Scripting
  - More information
    - Password synchronization
    - Agents account management

Managed systems

Description

Once the server plugins and loaded into Soffid, agents can be configured in order to synchronize repository information.

Addons installation

In order to configure a new agent, before you must downloaded and installed the connector whom includes this agent.

You could see the complete list of Synchronization Server Connectors in the following link: 3. Synchronization Server Connectors

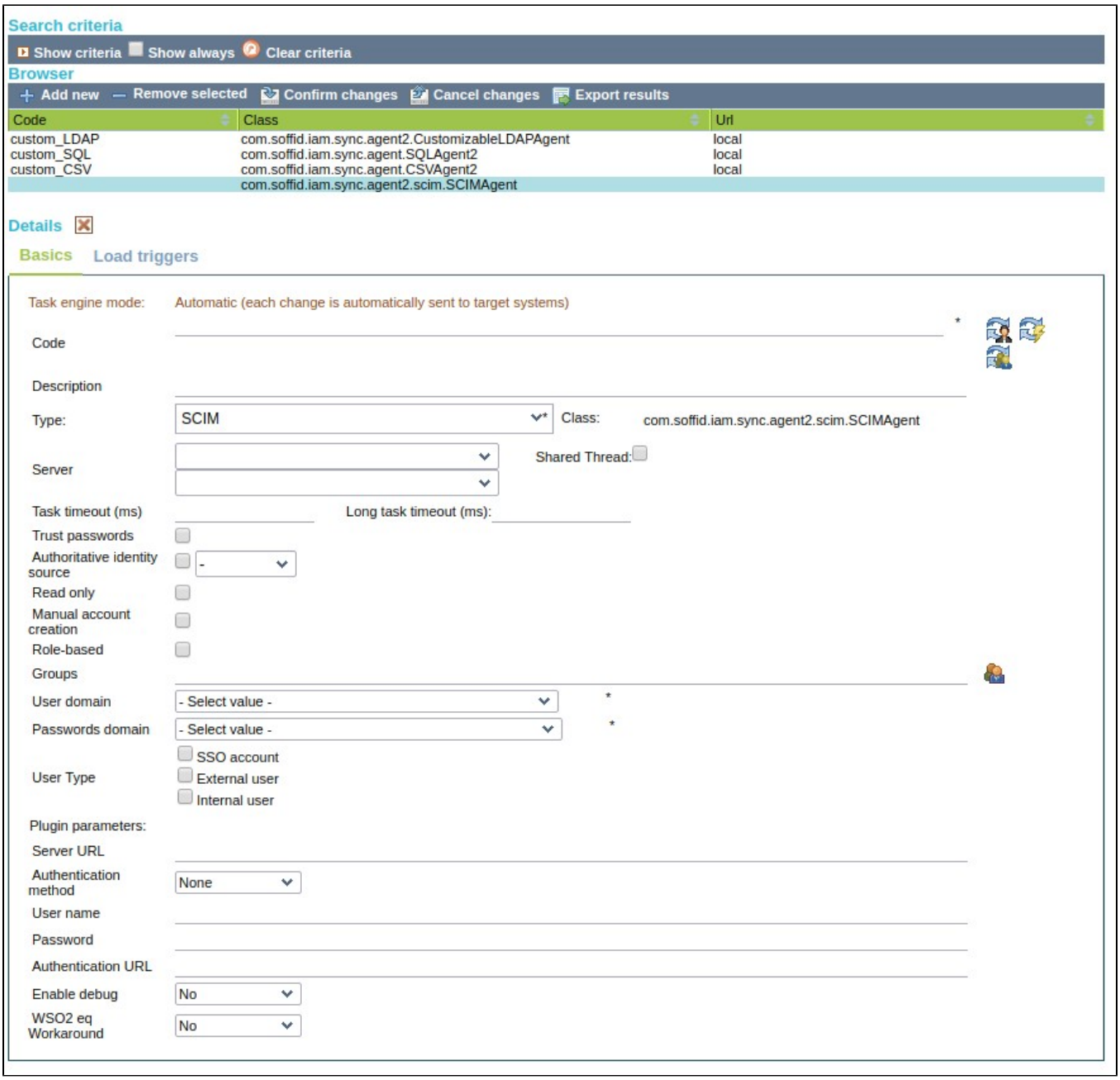
> For download and install one addon you could review our generic documentation about this process: Addons installation

Configuration

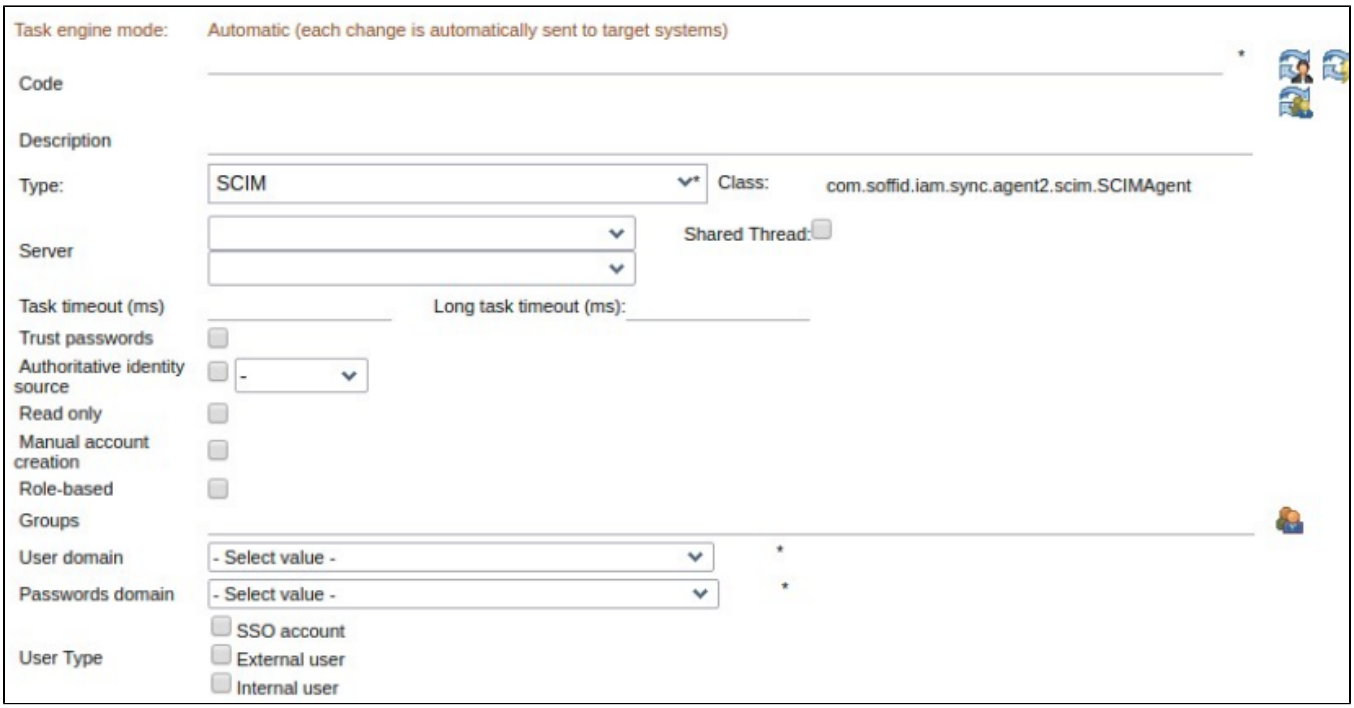
To create or update an agent, you may follow the next configuration data.

Not all the agents have all the configuration implemented, some agent only have some features available.

Basics



Generic parameters

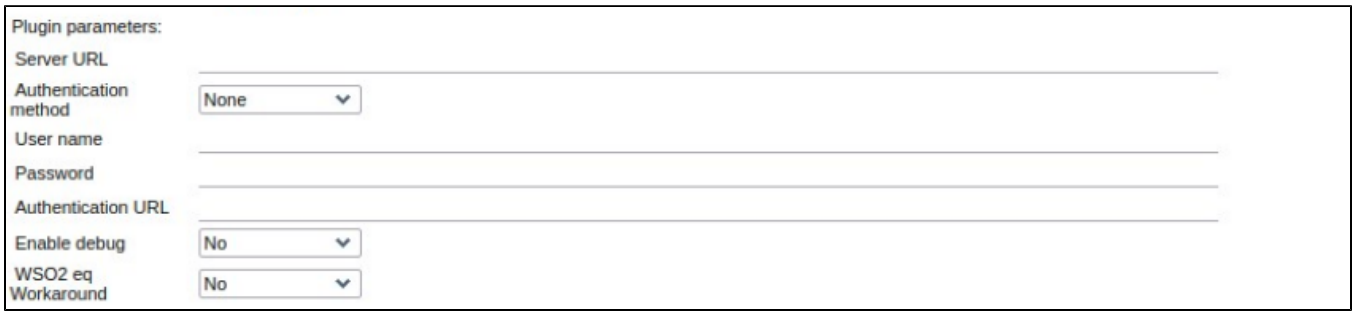


Each agent will have the following parameters:

Parameter	Description
Task engine mode	This is a info alert to show you the current "Task engine" configuration . For more information: Task engine
Code	Name of the agent
Description	Description of the agent
Type	Implementation of the server plugins included in the connectors installed
Server	The server where to run the agent: • If "*Each main synchronization server*" is selected, the agent will be run by every sync server • If you select a single sync-server, the agent only will be run in that server • If you leave the list in blank, the agent will be disabled
Shared Thread	To share the same thread to several synchronization servers
Task timeout (ms)	To add a timeout to the synchronization server tasks (query, insert, update, delete, update password, etc). If you add a timeout, when the connection get this timeout, the synchronization server will stop the request and add it to the queue for a new retry later.
Long task timeout (ms)	To add a timeout to the reconciliation server tasks (user, group, role, account, grants, etc). If you add a timeout, when the connection get this timeout, the synchronization server will stop the request (no retry is added).
Trust password	Check it if you can trust on it to propagate their passwords to Soffid. Trusted password agents differ from the non-trusted in: • Temporary passwords generated from the console only propagate to agents that have trusted password checked. In the other case, the agents only receive definitive passwords. • When a password has reached its expiry date it will automatically be disabled on agents where trusted password is not checked, so the user can no longer access it. • When the managed system detects a change in the user request password, the password will be propagated to Soffid only if in the agent associated trusted password is checked.
Authoritative identity source	Check this box if the agent will be used as the source for users information. Optionally, you can select a custom workflow to process incoming changes. User automatic task management page to schedule import tasks
Read only	If this box is checked, no change will be applied to the managed system. Only read operations will be allowed

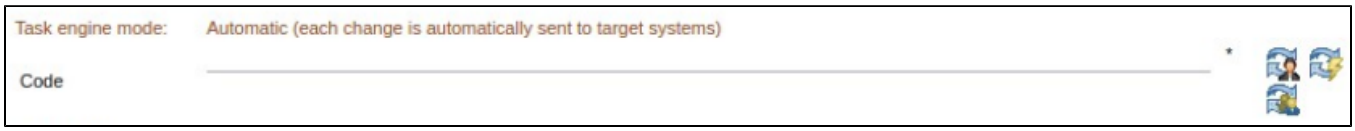
Manual accounts creation	Check it if you don't want Soffid to create automatically new accounts for the user
Role based	Check it if only users with any role on this agent should be created. Uncheck to allow users with no role on it
Groups	If any is specified, only users belonging to such organization unit will be created. Other users will be deleted or disabled
User and password domain	Selects the way accounts and password will be managed. See Agents account management page
User types	Only users of this type will be created. Any change made in this field involves all accounts to be recalculated. New ones will be added to the repository and managed systems. Some accounts will get disabled if the owner user does not longer belong to any authorized user type

Custom attributes



The other attributes depend on the used plugin. See Agents Guide to get details about specific plugin parameters.

Synchronization buttons



At the upper right side of the page, you will find three icons. This icons allows administrator to enforce synchronization of users,  , roles,  and groups,  . If you press on them, a set of tasks will be scheduled to synchronize all of them.

It is important to clearly specify the groups and the users type that should be spread to the managed system. Any user that does not belong to these groups or user types will be automatically deleted or disabled from the managed system. As an exception, if no group is entered, all groups will be spread.

Regarding the account names to be used in the managed system, the users domain must be specified too. The users domain defines the rule by which the account name is generated based on the user name and its attributes. If the account name is the same as the user name (as it is normally the case), the “Default user domain” should be used.

Regarding the password domain, the user will share the same password for all the managed systems with the same password domain. There is “Default password domain” that will share the passwords used on Soffid console.

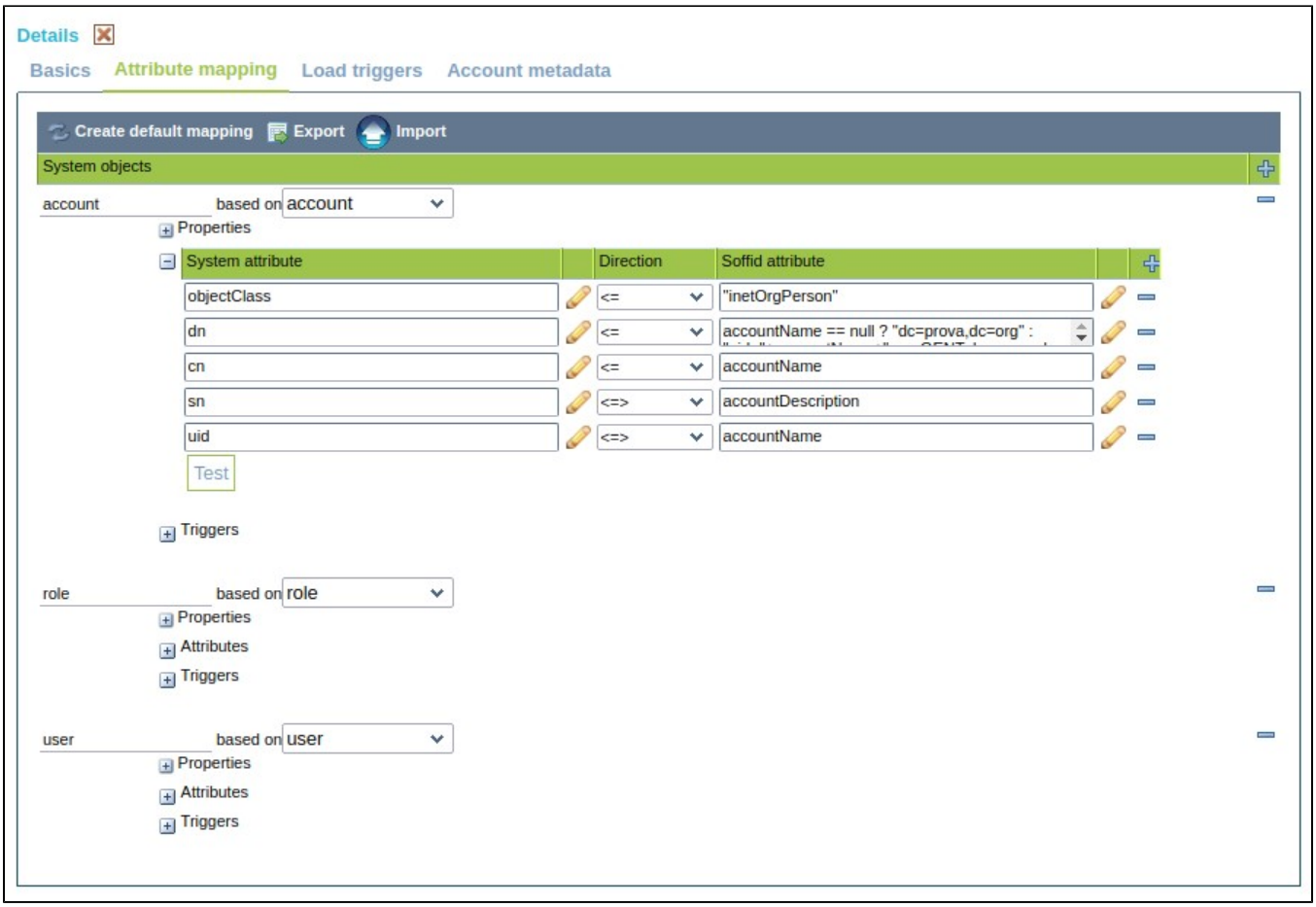
On the right hand side of the agent code there are two icons that permit spreading all users or roles to the managed system.

Attribute Mapping

Soffid administrator have the chance to easily customize attribute mappings without having to code it using Java.

When a agent allows this kind of customization, a new tab named "Attribute mapping" will appear. On this tab, the user can make either inbound or outbound attribute mappings. The left hand side attributes are managed system attributes, so they are agent dependent. The right side attributes are Soffid attributes and must be selected from the list below.

When the mapping is bidirectional, both sides of the mapping must be naming a single account, but when the mapping is one way, the source attribute can be replaced with a bean Shell expression.



Properties

Some agents require to configure some custom attributes in this properties section.

You could see these attributes in the "*Properties*" section of each single page

Attributes

System attributes

A configuration agent must define objects types that can be created on it. Each object mapping defines an agent object name and a bound Soffid object type. For each mapped object, a list of attributes will be displayed. At the left column the system's attribute name will be displayed, and at the rightmost column, the Soffid's attribute name will be shown.

Directions

At the center column, an arrow will show the direction the information flows. When the information flows from the system (left), to Soffid (right), the left column name can be replaced by a bean shell expression. This expression will be evaluated on the system object prior to uploading it to Soffid.

When the information flows from Soffid (right) to the managed system (left), the right column can contain a bean shell expression that will be evaluated prior to provisioning the user. Here are some examples:

System attribute	Direction	Soffid attribute	Meaning
cn	<=>	accountName	The account name is the CN attribute of the LDAP

departmentNumber	<=	for (group: secondaryGroups) {
		
		if (group.get("name").equals (primaryGroup)) {
		
		return group.get("description");
		
		}
		
		}
		
		return null;
			Assigns the group description of the primary group to the departmentNumber attribute

baseDN	<=	"ou="+primaryGroup+",dc=soffid,dc=org"	Assigns the base dn of the user to the proper organization unit that is below dc=soffd, dc=org.

Soffid attributes

List of Soffid attributes are described below

- User Object
- Account Object
- Group Object
- Role Object
- Grant Object
- Maillist Object
- Membership Object

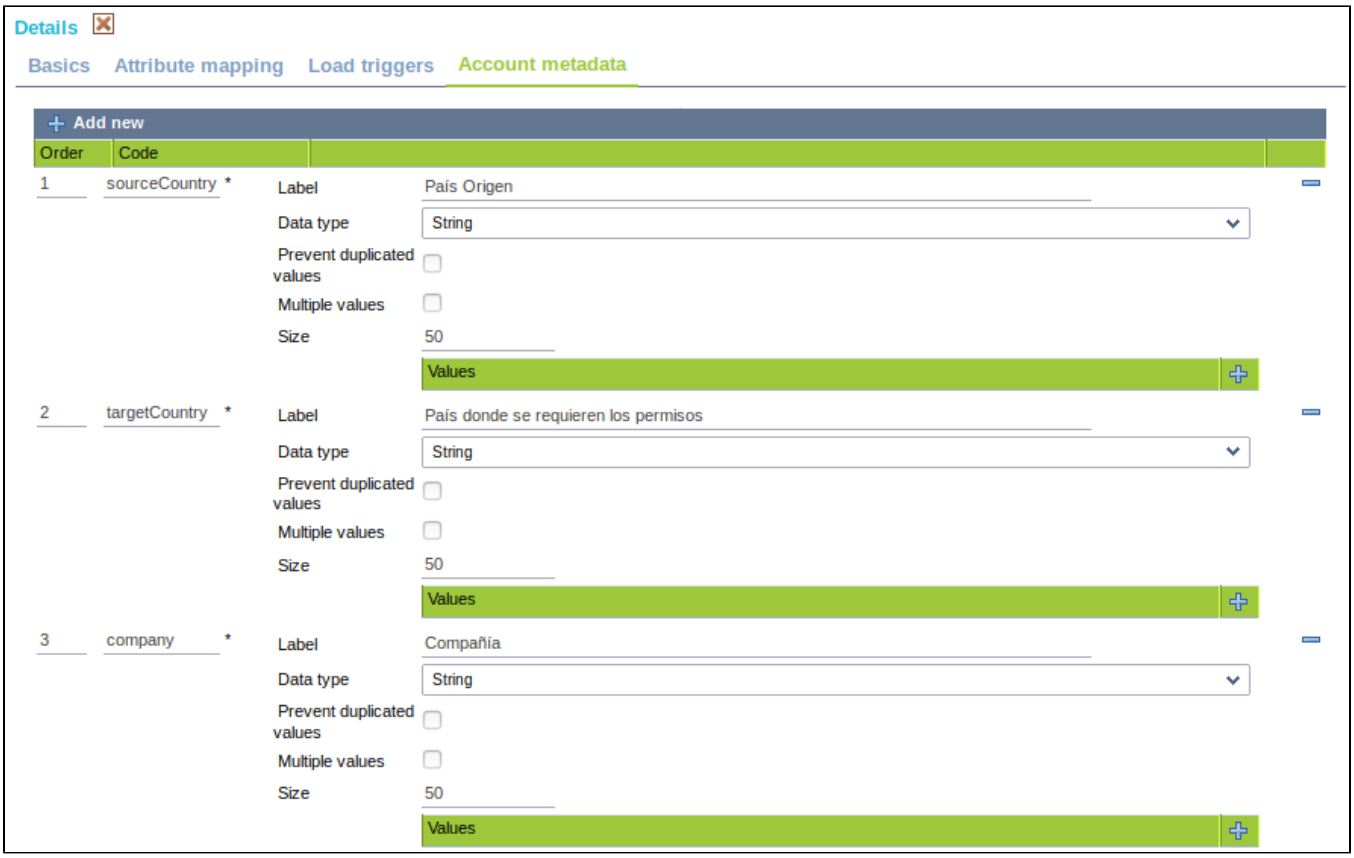
When evaluatting any expression, either the system or soffid attributes are available as script variables. More over, the following variables are available:

Variable	Content
serverService	Server API that enables an easy object query [Search the link "Public API Module" or "Data & Service model"]
serviceLocator	Spring Singleton that gets access to any published service bean. Only availble on main syncserver
remoteServiceLocator	Singleton that gets access to any remote published service bean.
THIS	HashMap that contains any soffid or system managed attribute. It can be used when the attribute name is not a valida java identifier.
dispatcherService	Service that allows the script to get or update information in target system. See more here

Load triggers

Account metadata

Accounts are default objects in Soffid but, depending on the system they link to, they can not be treated as an identity, group, role, mailing list or application, so custom objects to add additional data to the accounts should be created specifically for each agent using the account metadata tab.



To proceed to it, click on + Add new to set a new metadata.

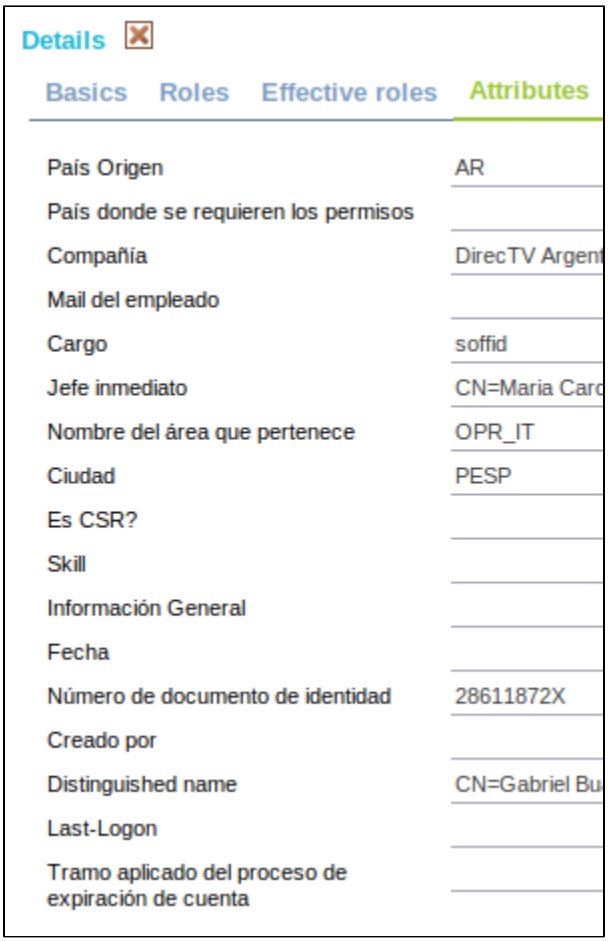
At this point, the following should be indicated:

1. Order: number that determines the order in which the custom object will appear.
2. Code: text used by the system to refer to the object.
3. Label: description understandable to the user.
4. Data type: at this point, the data type of this metadata must be selected. The data type includes the default ones as well as the custom objects that may have been created in the system.
5. Prevent duplicated values: if this flag is enabled, there can not be an identical duplicate metadata.
6. Multiple values: if this flag is enabled, the metadata may contain more than one value.
7. Size: set the max length.

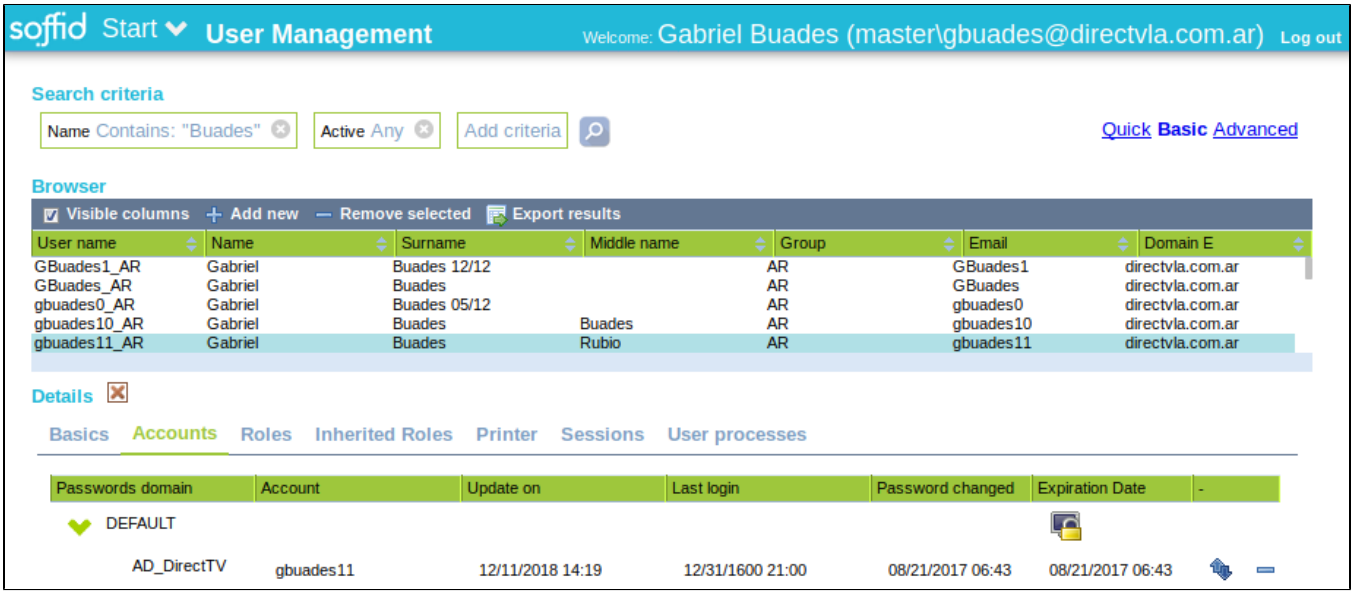
Once completed, click on Confirm changes to save data.

If the created metadata should be deleted, click on , located on the right of the created metadata, and then on Confirm changes to save data.

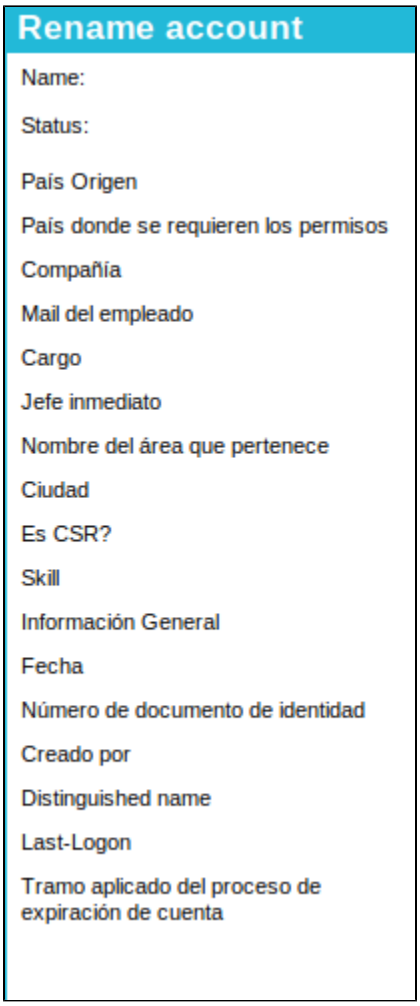
Now, once configured the account metadata in the corresponding agent, access the accounts screen, select one of the system to which that agent points and go to "accounts" tab. There it is possible to see that the metadata are sorted as was indicated in the agent.



It can also be checked on the "user" screen, by selecting a user and accessing the "Accounts" tab



Then, click on  located to the right of the account pointed to by the agent in which the metadata were defined.



A window opens and shows the account metadata, sorted how they were set in the agent.

Scripting

In the agents configuration it may be possible use scripting to include logic in the attribute mappings and in the trigger scripts.

In the attribute mapping, if you use a script in one side, it is mandatory a single direction to the other side:

- System attribute <= script
- script => Soffid attribute

Below a easy script to send a full name to the system:

```
system attribute <= return firstName + lastName;
```

Below a more complex script to create a main domain if it doesn't exist in Soffid:

```
String mailDomain = null;
if (email != void && email != null && email.contains("@")) {
    String[] mailTokens = email.split("@");
    mailDomain = mailTokens[1];
}
com.soffid.iam.service.MailListsService service = com.soffid.iam.ServiceLocator.instance().
getMailListsService();
com.soffid.iam.api.MailDomain domain = service.findMailDomainByName(mailDomain);
if (domain==null) {
    domain = new com.soffid.iam.api.MailDomain();
    domain.setCode(mailDomain);
    domain.setDescription(mailDomain);
    domain.setObsolete(new Boolean(false));
    domain = service.create(domain);
}
return mailDomain;

=> mailDomain
```

Below you could find the API for the internal classes of Soffid:[Search the link "Public API Module" or "Data & Service model"]

Below you could find a set of sample scripts: Sample scripts

Below you could find a link with the SCIM Query Language used in some methods as findUserByJsonQuery("query"): 5. SCIM filter language

Below you could find a set of custom utility classes: Utility classes

More information

Password synchronization

The passwords a user have on an agent will be synchronized with any other "single user account" the user has on this agent. Shared accounts will never get its password synchronized.

Password in an agent will be also synchronized with any other account the user has on other agents that are sharing the same password domain.

The password change can be produced by an operator using Soffid console, the user itself using Soffid Self Service portal or a timed automatic task. Furthermore, some managed systems can forward their password to Soffid in order to get them synchronized. In order to accept this password changes coming from managed systems, the trusted passwords box must be checked for the source agent.

Mind that this is the flow for normal user passwords. Temporary passwords generated by Soffid console will only be sent to agents marked as trusted. Agents not checked as trusted will have a random new password instead. Later, when the user changes the password on Soffid or any trusted system, the new password will be notified to Soffid by the managed system, and every agent on the same password domain will actually get the new password.

Agents account management

The agent configuration sets the way accounts are created and disabled.

Whenever a user is modified, the following rules will be applied to check if the user should have or not an account on this agent:

1. The user type is check against valid user types
2. If there is a business unit or group bound to the agent, the user memership will be assessed.
3. If the role based box is checked, the system will verify if the user has any role or entitlment assigned on this agent.

If the user does not apply for any of the conditions, every account the user has at this agent will be changed to Disabled status.

If the user verifies avery one of the conditions, the user can have an account on this agent. Every account the uesr has at this agent will be changed to Enabled status.

Unless the "manual account creation" is checked, if the user can have an account on this agent, but it has no one, the account creation method will be invoked. To create it, Soffid will search for the user domain bound to this agent and will follow its configuration. If the user domain is configured with a script, this script will be executed and the result value will be accepted as the new account name. Mind that if the script returns a null value, no account can be created.

If the returning value from the script clashes with an existing account, the existing account will remain unchanged, unless the existing account is marked as a unmanaged account. In such a case, the account will be changed from unmanaged state to single user.