LDAP Connector

LDAP Connector

Introduction

Description

This connector implements the LDAP standard and it is used to connect the Sync-Server with every server that allows this communication protocol.

Managed System

There are a lot of servers and products that use this standard, for instance, the most known systems are:

- 389 Directory Server.
- Apache Directory Server.
- OpenLDAP.
- OpenDJ.
- Active Directory.
- Oracle Directory Server.

For more information: List of LDAP software.

> If your system is not in the previous list, it's possible to include it easily!
>
> For more information to check if your system may be synchronized with this connector you do not hesitate to contact us through our Contact form

Prerequisites

It is needed a user with full administrator access.

Download and Install

This addon is located in the Connectors section and its name is **LDAP plugin.**

> For more information about the installation process you can visit the Addons Getting started page.

Agent Configuration

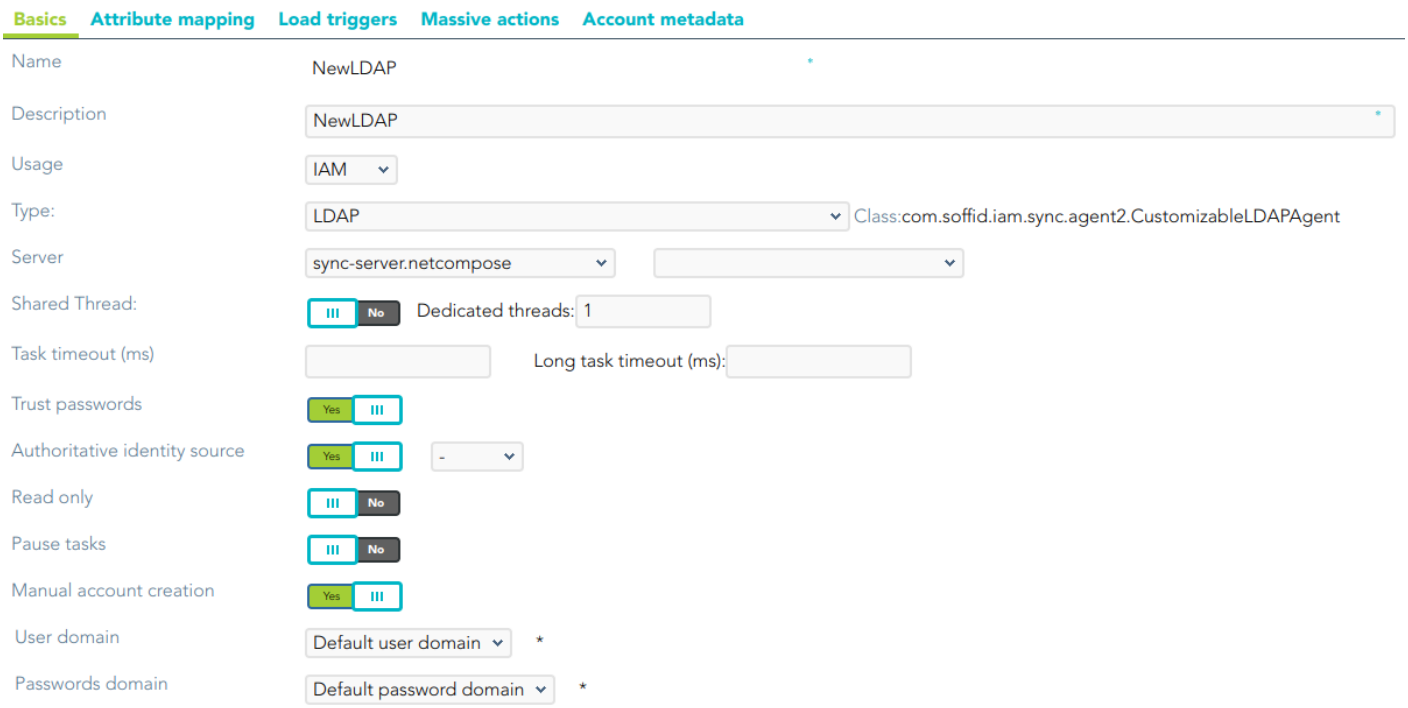
Basic

Generic parameters

After the installation of the addon, you may create and configure agent instances.

To configure this LDAP Connector you must select "LDAP-Custom (with triggers)" in the attribute "Type" of the generic parameters section in the agent's page configuration.

> For more information about how you may configure the generic parameters of the agent, see the following link: Agents configuration



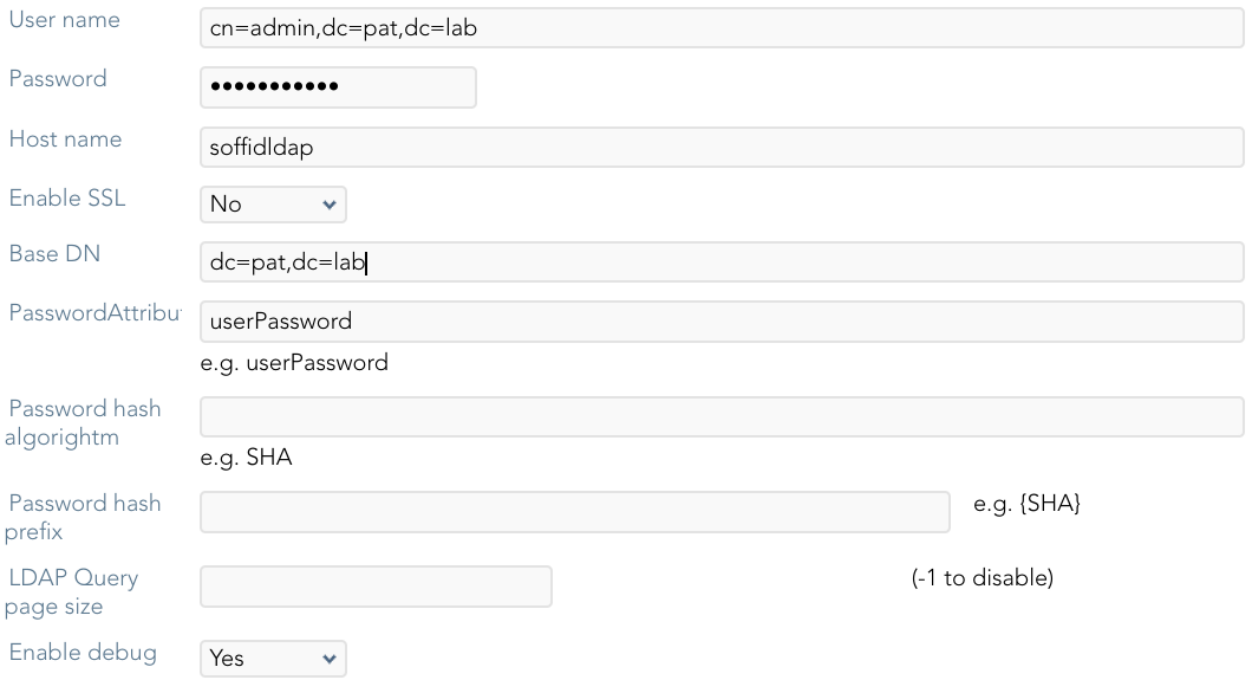
Custom parameters

Below there are the specific parameters for this agent implementation:

Parameter	Description
User name	User name in DN format, including base name if needed
Password	Password
Host name	Host name of the server
Enable SSL	
Base DN	LDAP Base name
PasswordAttribute	
Password hash algorithm	The algorithm is used to encrypt the password. For instance SHA-1, SHA-256, MD5, etc
Password hash prefix	
LDAP Query page size	
Enable debug	Two options: Yes, No. When it is enabled more log traces are printed in the Synchronization Server log

Attribute mapping

This connector can manage users, accounts, roles, groups, and grants.



> As a limitation, it cannot detect password changes to be propagated to other systems.

Properties

Some agents require to configure some custom attributes, you will use the properties section to do that.

Property	Value
rename	true
key	LDAP attribute where Soffid account name is stored. If the property is not present, object will be searched by its distinguishedName
modificationTimestamp	LDAP attribute
removeDisabledAccounts	Set to true to remove disabled accounts from LDAP server

If a key value is set, LDAP connector will search for objects based on this LDAP attribute value, rather than its DN. Thus, an index on this attributed is highly recommended.

Renaming

To support object renaming, Soffid needs to store the Soffid account name on a specific LDAP attribute. It's highly recommended to index such a field. To enable it, add the following properties to each object mapping. At any time, object renaming can be disabled by setting the property rename to false.

Property	Value
rename	true

Attributes

You can customize attribute mappings, you only need to select system objects and the Soffid objects related, manage their attributes, and make either inbound or outbound attribute mappings.

Using a windows connector you can map users, groups, and role objects. Active Directory membership is automatically managed based on user and group mappings.

You can map users, groups, and role objects. User membership must be managed on the role members' attribute expression.

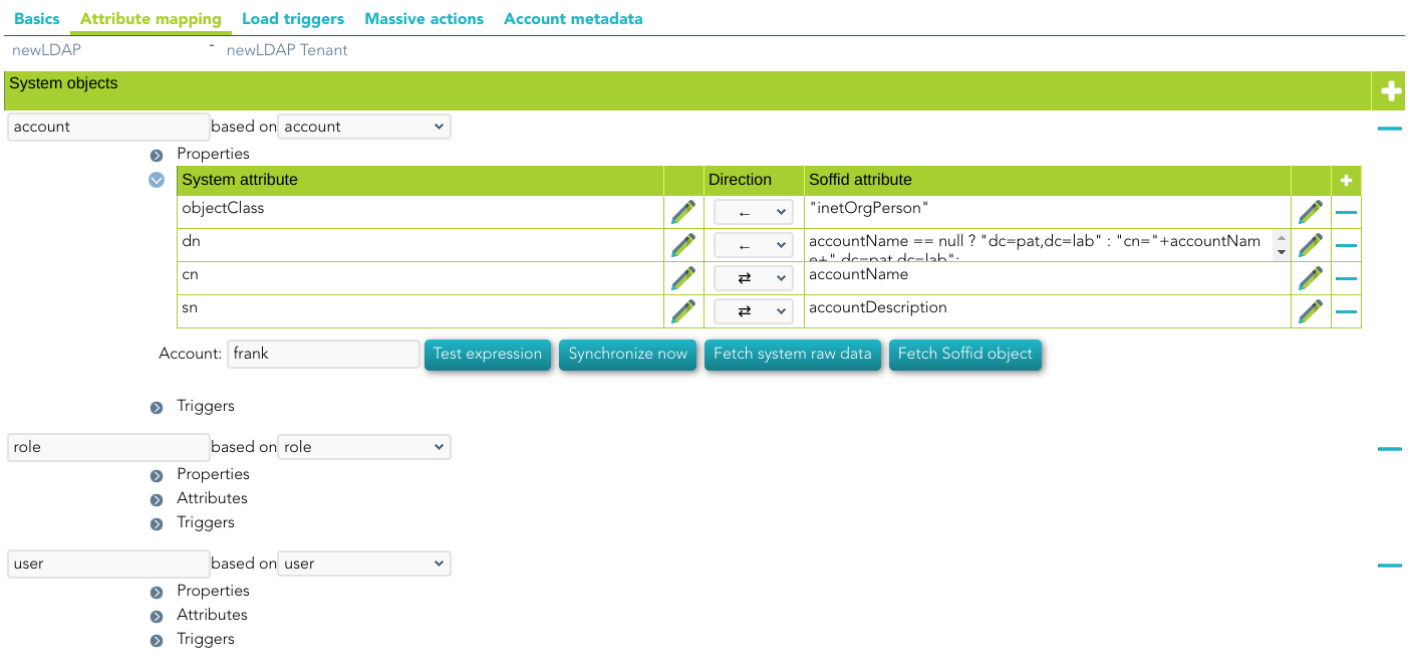
Any object mapping must have the following system attributes:

System attribute	Value
objectClass	LDAP Object Class. It can evaluate to an array of objects

System attribute	Value
dn	Full qualified object name

> For more information about how you may configure attribute mapping, see the following link: Soffid Attribute Mapping Reference

For instance:



Triggers

You can define BeanShell scripts that will be triggered when data is loaded into the target system (outgoing triggers). The trigger result will be a boolean value, true to continue or false to stop.

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

> To view some examples, visit the Outgoing triggers examples page.