

Configure CMS LDAP Integration

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Configure](#)

[Active Directory Server Settings](#)

[Import Settings](#)

[Field Mapping Expressions](#)

[Resilient/Scalable Deployment](#)

[Web Interface API](#)

[LDAP API Objects](#)

[LdapServers](#)

[LdapMappings](#)

[LdapSources](#)

[Migrating Web GUI Configurations to API](#)

[Step 1. Notating Web GUI Active Directory Settings](#)

[Step 2: Navigate to LDAP parameters within API](#)

[Step 3. Create ldapServer within API](#)

[Step 4. Create ldapMappings within API](#)

[Step 5. Create ldapSources within API](#)

[Step 6. Verify Settings Change Through ldapSync](#)

[Verify](#)

[Troubleshoot](#)

Introduction

This document describes the step-by-step process of integrating Lightweight Directory Access Protocol (LDAP) with the Cisco Meeting Server (CMS).

Prerequisites

Requirements

Cisco recommends you have knowledge of these topics:

- CMS Callbridge version 2.9 or later
- Microsoft Lightweight Directory Access Protocol (LDAP)


Components Used

The information in this document is based on CMS 3.0.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

This document focuses on a number of topics that deal with LDAP integration with the CMS. It also includes steps on how to migrate active directory configurations from the CMS GUI in **Configuration > Active Directory** to API.

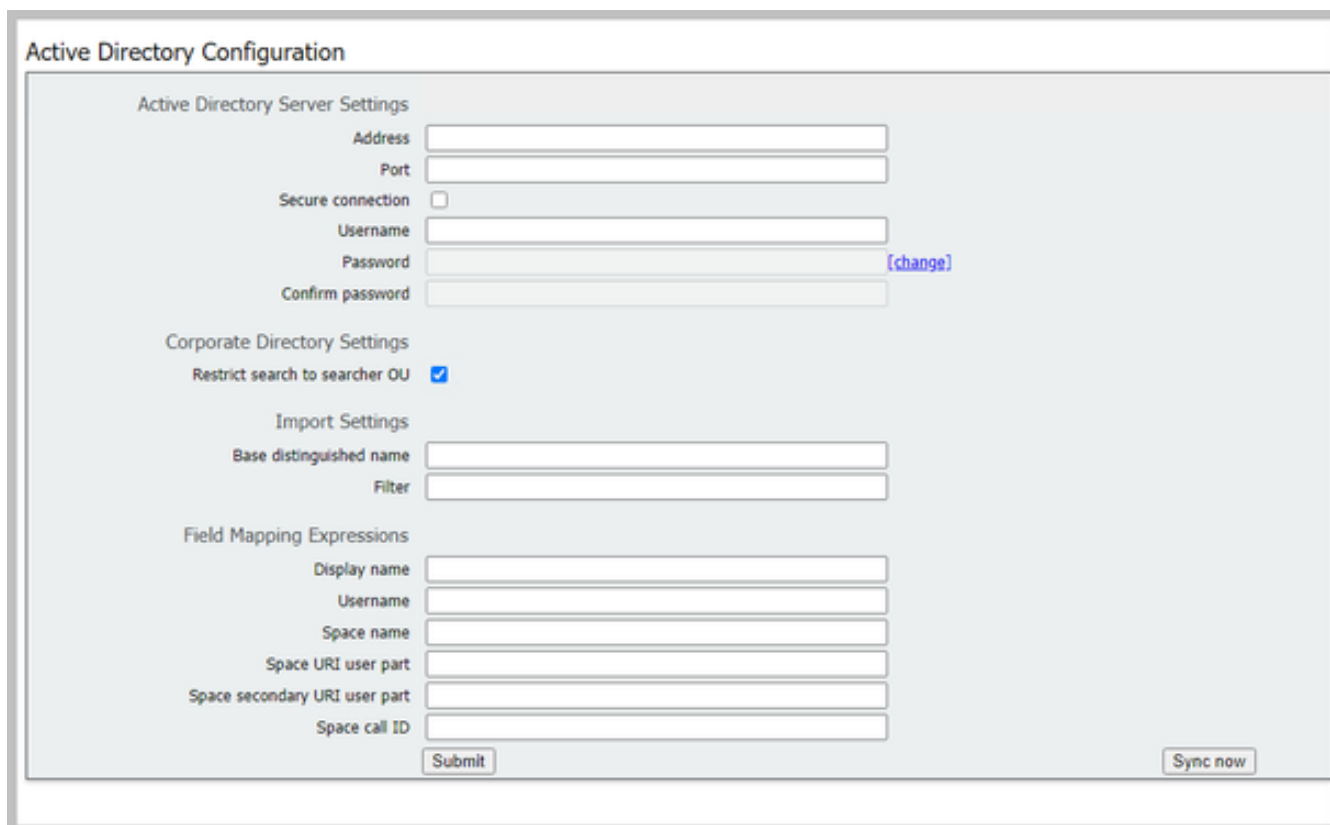
 **Note:** The only LDAP servers that are supported for CMS are Microsoft Active Directory, OpenLDAP, Directory LDAP3, and Oracle Internet Directory.

 **Note:** LDAP Configurations in the web GUI could be removed in future releases of CMS.

Configure

The only scenario that you would configure the LDAP configuration within the Web Interface is if you have a single LDAP source to import into CMS.

 **Note:** Active Directory could be removed from the Web GUI in the later releases of CMS.



The screenshot shows the 'Active Directory Configuration' web interface. It is organized into several sections:

- Active Directory Server Settings:** Includes input fields for Address, Port, Username, Password (with a [change] link), and Confirm password. There is an unchecked checkbox for 'Secure connection'.
- Corporate Directory Settings:** Includes a checked checkbox for 'Restrict search to searcher OU'.
- Import Settings:** Includes input fields for 'Base distinguished name' and 'Filter'.
- Field Mapping Expressions:** Includes input fields for 'Display name', 'Username', 'Space name', 'Space URI user part', 'Space secondary URI user part', and 'Space call ID'.

At the bottom of the form, there are two buttons: 'Submit' and 'Sync now'.

Active Directory Server Settings

Configure the connection to the LDAP Server with:

Address	This is the hostname or IP address of your LDAP server.
Port	389 for Unsecure & 636 for secure connection (must check the secure connection check box)
Username	The Distinguished Name (DN) of a registered user. You could create a user specifically for this purpose. Example: cn=Tyler Evans,cn=Users,OU=Engineering,dc=YourCompany,dc=com
Password	The password for the user name you are using
Secure Connection	Check this box if using port 636

Import Settings

Import Settings is used to control which users are imported:

Based Distinguished Name	The node in the LDAP tree from which to import users. This example is a sensible choice for base DN to import users	Example: cn=Users,dc=sales,dc=YourCompany,dc=com
Filter	a filter expression that must be satisfied by the attribute values in a users LDAP record. The syntax for the Filter field is described in rfc4515.	Example: mail=*

Field Mapping Expressions

The field mapping expressions control how the field values in the Meeting Server user records are constructed from those in the corresponding LDAP records.

Display Name
User Name
Space Name
Space URI User Part

Secondary Space URI User Part
Space Call ID

Resilient/Scalable Deployment

There are two scenarios where you would need to configure LDAP within the API. One scenario is when you have a clustered deployment of 3 or more nodes and the second scenario is when you have more than one LDAP source to import users from.

Web Interface API

Navigate to the API Web Interface by logging into your Web Admin of your **CMS > Configuration > API**. Here is where you make all of your API configurations.

LDAP API Objects

After navigating to the API, type "**ldap**" in the filter bar to display all of the LDAP configurations you can make.

API objects

This page shows a list of the objects supported by the API. Where you see a ▶ control, you can expand that section to either show a list of objects of that specific type or the details of one specific section of configuration.

Filter (10 of 116 nodes)

```

/api/v1/ldapMappings ▶
/api/v1/ldapMappings/<id>
/api/v1/ldapServers ▶
/api/v1/ldapServers/<id>
/api/v1/ldapSources ▶
/api/v1/ldapSources/<id>
/api/v1/ldapSyncs ▶
/api/v1/ldapSyncs/<id>
/api/v1/ldapUserCoSpaceTemplateSources ▶
/api/v1/ldapUserCoSpaceTemplateSources/<id>

```

Objects in the hierarchy that reside in the “/ldapMappings”, “/ldapServers” and “/ldapSources” nodes in the object tree relate to the Meeting Servers interaction with one or more LDAP servers (for instance, Active Directory) which are used to import user accounts to the Cisco Meeting Server.

LdapServers

One or more LDAP servers must be configured, with each one having associated username and password information for the Meeting Server to use to connect to it for the purpose of retrieving user account information from it.

[« return to object list](#)

/api/v1/ldapServers

address *	<input type="checkbox"/>	<input type="text"/>	- required
name	<input type="checkbox"/>	<input type="text"/>	
portNumber *	<input type="checkbox"/>	<input type="text"/>	- required
username	<input type="checkbox"/>	<input type="text"/>	
password	<input type="checkbox"/>	<input type="text"/>	
secure *	<input type="checkbox"/>	true ▼	- required
usePagedResults	<input type="checkbox"/>	<unset> ▼	
<input type="button" value="Create"/>			

* = Required

Address*	address of the LDAP server to connect to
----------	--

Name	associated name (from version 2.9 onwards)
portNumber *	Port 389(unsecure) or Port 636(Secure)
Username	username to use when retrieving information from the LDAP server
Password	password of the account associated with username
Secure *	whether to make a secure connection to the LDAP server. If “true” then TLS is used; if “false”, TCP is used.
usePagedResults	whether to use the LDAP paged results control in search operations during LDAP sync; if not set, the paged results control is used. Oracle Internet Directory requires this parameter to be set to “false” (from version 2.1).

LdapMappings

One or more LDAP mappings are also required, which define the form of the user account names which are added to the system when users are imported from configured LDAP servers.

[« return to object list](#)

/api/v1/ldapMappings

jidMapping	<input type="checkbox"/>	<input type="text"/>
nameMapping	<input type="checkbox"/>	<input type="text"/>
cdrTagMapping	<input type="checkbox"/>	<input type="text"/>
coSpaceUriMapping	<input type="checkbox"/>	<input type="text"/>
coSpaceSecondaryUriMapping	<input type="checkbox"/>	<input type="text"/>
coSpaceNameMapping	<input type="checkbox"/>	<input type="text"/>
coSpaceCallIdMapping	<input type="checkbox"/>	<input type="text"/>
authenticationIdMapping	<input type="checkbox"/>	<input type="text"/>
		<input type="button" value="Create"/>

* = *Required*

jidMapping*	The template for generating user JIDs from the associated LDAP servers entries, for instance \$sAMAccountName\$@example.com. Note: user JIDs generated by jidMapping are also used as URIs so must be unique and not the same as any URI or call ID.
nameMapping	The template for generating user names from the associated LDAP servers entries; for instance “\$cn\$” to use the common name.
cdrTagMapping	The template for generating a users' cdrTag value. Can be set either to a fixed value or be constructed from other LDAP fields for that user. The user’s cdrTag is used in

	callLegStart CDRs. See the Cisco Meeting Server CDR Reference for details.
coSpaceUriMapping	If these parameters are supplied, they ensure that each user account generated by this LDAP mapping has an associated personal coSpace.
coSpaceSecondaryUriMapping	For that coSpace to be set up as required, these parameters provide the template for setting the coSpaces' URI, displayed name and configured Call ID. For example, setting coSpaceNameMapping to "\$cn\$ personal coSpace" ensures that each user's coSpace is labelled with their name followed by "personal coSpace".
coSpaceNameMapping	
coSpaceCallIdMapping	
authenticationIdMapping	The template for generating authentication IDs from the associated LDAP servers entries, for instance "\$userPrincipalName\$"

LdapSources

A set of LDAP sources then need to be configured, which tie together configured LDAP servers and LDAP mappings, along with parameters of its own, which correspond to the actual import of a set of users. An LDAP source takes an LDAP server / LDAP mapping combination and imports a filtered set of users from that LDAP server. This filter is determined by the LDAP sources "baseDn" (the node of the LDAP servers tree under which the users can be found) and a filter to ensure that user accounts are only created for LDAP objects that match a specific pattern.

Status ▾
Configuration ▾
Logs ▾

[« return to object list](#)

/api/v1/ldapSources

server *	<input type="checkbox"/>	<input type="text"/>	<input type="button" value="Choose"/>	- required
mapping *	<input type="checkbox"/>	<input type="text"/>	<input type="button" value="Choose"/>	- required
baseDn *	<input type="checkbox"/>	<input type="text"/>		- required
filter	<input type="checkbox"/>	<input type="text"/>		
tenant	<input type="checkbox"/>	<input type="text"/>	<input type="button" value="Choose"/>	
userProfile	<input type="checkbox"/>	<input type="text"/>	<input type="button" value="Choose"/>	
nonMemberAccess	<input type="checkbox"/>	<unset> ▾		
<input type="button" value="Create"/>				

* = Required

server*	The ID of a previously-configured LDAP server
mapping*	The ID of a previously-configured

	LDAP mapping (
baseDn*	The distinguished name of the node in the LDAP servers tree from which users are to be imported from, for instance “cn=Users,dc=,dc=com”
filter	
tenant	
userProfile	
nonMemberAccess	

Migrating Web GUI Configurations to API

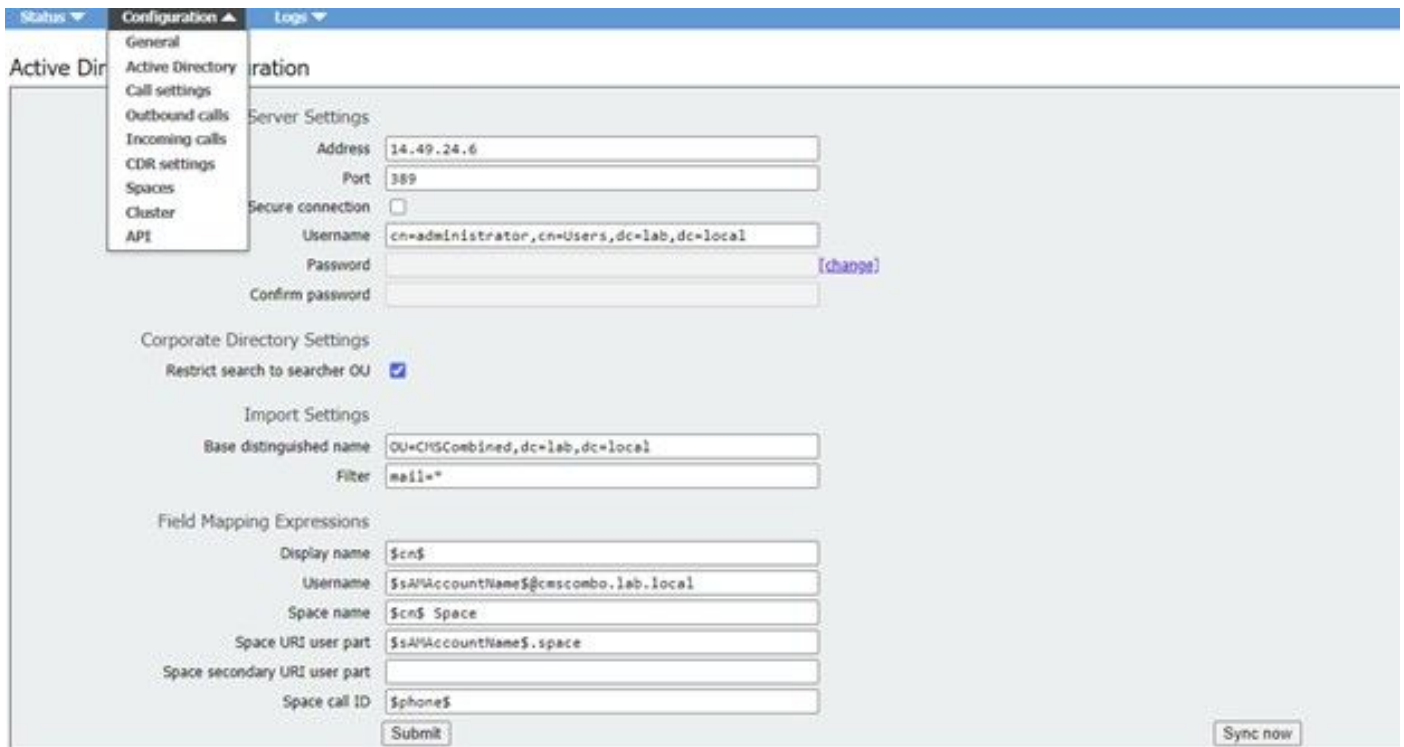
This section discusses how you migrate LDAP web GUI configurations to API. If you currently have Ldap configurations in the web GUI and you want to migrate this information to API then use this example to avoid losing data.



Note: What happens when you move AD from GUI to API? If you configure the API first before removing the GUI Active directory settings, the user information remains unchanged; call ID and secret also remain the same. However, If you remove the GUI before configuring the API afterwards, new call ID and secrets are assigned to users.

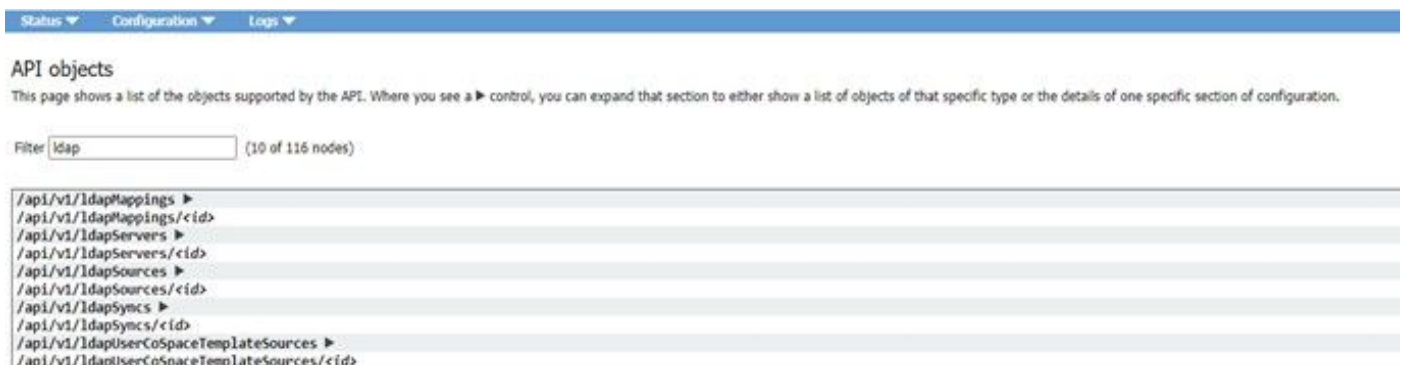
Step 1. Notating Web GUI Active Directory Settings

Navigate to **Configurations > Active Directory** to see the LDAP configurations for your Web GUI. Take a screen shot of this or copy and paste these contents into a text editor to be used later.



Step 2: Navigate to LDAP parameters within API

Navigate to **Configurations > API >** Type “**Ldap**” in the filter bar.



Displayed is a list of LDAP configurations.

Step 3. Create ldapServer within API

From this list, click on **ldapServers** and then select “**Create New**”. Refer to the screen shot or text editor for the contents that were within your web GUI Active Directory. You are now going to copy the “**Active Directory Server Settings**” from the Web Gui into the corresponding API configurations.



Step 4. Create ldapMappings within API

After you complete Step 4., navigate to ldapMapping within the API. **Configurations > API > Filter “ldapMapping”** and click **Create New**.

/api/v1/ldapMappings

jidMapping	<input type="checkbox"/>	
nameMapping	<input type="checkbox"/>	
cdrTagMapping	<input type="checkbox"/>	
coSpaceUriMapping	<input type="checkbox"/>	
coSpaceSecondaryUriMapping	<input type="checkbox"/>	
coSpaceNameMapping	<input type="checkbox"/>	
coSpaceCallIdMapping	<input type="checkbox"/>	
authenticationIdMapping	<input type="checkbox"/>	
		<input type="button" value="Create"/>

/api/v1/ldapMappings

jidMapping	<input checked="" type="checkbox"/>	\$sAMAccountName\$@lab.local
nameMapping	<input checked="" type="checkbox"/>	\$cn\$
cdrTagMapping	<input type="checkbox"/>	
coSpaceUriMapping	<input checked="" type="checkbox"/>	\$sAMAccountName\$.cs
coSpaceSecondaryUriMapping	<input checked="" type="checkbox"/>	*\$sipPhone\$
coSpaceNameMapping	<input checked="" type="checkbox"/>	\$sAMAccountName\$'s Space
coSpaceCallIdMapping	<input type="checkbox"/>	
authenticationIdMapping	<input type="checkbox"/>	
		<input type="button" value="Create"/>

Copy the Field Mapping Expressions from the Web GUI from **Configurations > Active Directory > Field Mapping Expressions**. Next, navigate to **Configuration > API > filter “ldapmapping”** and then click **Create**.

Field Mapping Expressions (Web GUI)	API
Display name	nameMapping
Username	jidMapping
Space name	
Space URI user part	coSpaceURIMapping
Space secondary URI User Part	coSpaceSecondaryUriMapping
Space call ID	

Step 5. Create ldapSources within API

Now migrate the Corporate Directory/Import settings from the Web GUI into the LDAP Sources API configurations, **Configuration > API > filter “ldapSources”** and click the arrow next to **LdapSources** and then select **create new**.

/api/v1/ldapSources

server * Choose - required

mapping * Choose - required

baseDn * - required

filter

tenant Choose

userProfile Choose

nonMemberAccess <unset> v

Create

Select the LDAP Mapping and LDAP server that you configured in Steps 3. and 4.

/api/v1/ldapSources

server * 19780856-00ec-4e40-a197-58958718f356 Choose - required

mapping * af64add8-0273-4779-8652-01b46b30e7e6 Choose - required

baseDn * OU=CMSCombined,dc=lab,dc=local - required

filter mail=

tenant Choose

userProfile Choose

nonMemberAccess <unset> v

Create

Select the LDAP Mapping and LDAP server you just configured and then add the baseDN and filter from the Web Gui to the API configuration.

Import Settings (Web Gui)	API LdapSource
Base Distinguished Name	baseDn
Filter	filter

Step 6. Verify Settings Change Through ldapSync

You can now confirm that it works. Navigate to ldapSyncs in API, **Configuration > API > filter ‘ldapSyncs’** and click it and select **Create New**.

You do not have to fill anything out, just select **Create**. This begins the sync process. After 30 secs – 1 min, refresh the page in order to verify that you get a complete status and a 200 OK returned.

Verify

Ensure that all fields are properly configured.

Troubleshoot

There is currently no specific troubleshooting information available for this configuration.