PNMsoft Knowledge Base

Sequence Admin Guides

Active Directory Synchronization

March 2015

Product Version 7.x and above

© 2016 PNMsoft All Rights Reserved

This document, including any supporting materials, is owned by PNMsoft Ltd and/or its affiliates and is for the sole use of the PNMsoft customers, PNMsoft official business partners, or other authorized recipients. This document may contain information that is confidential, proprietary or otherwise legally protected, and it may not be further copied, distributed or publicly displayed without the express written permission of PNMsoft Ltd. or its affiliates.

PNMsoft UK 38 Clarendon Road Watford Hertfordshire WD17 1JJ

Tel: +44(0)192 381 3420 • Email: info@pnmsoft.com • Website: www.pnmsoft.com

Microsoft Partner
Gold Application Development

Table of Contents

General Document Information

Purpose

This document is designed to instruct you on how to configure the Active Directory Synchronization Service after installing Sequence for importing users from your Active Directory/Directories into Sequence. The Active Directory Synchronization Service will only pull information from your Active Directory and will never write to it.

Prerequisites

- Read access to the organization's active directory.
- Administrative access to the Sequence Administration environment (i.e. global administrator).
- Knowledge of LDAP query and filter.
- Access to the Windows Services console on the server(s) where the Active Directory Synchronization Service is installed.

About the Active Directory Synchronization Service

The Active Directory Synchronization Service (ADSS) is a Windows service which allows easier user management within Sequence. The ADSS enables importing objects from your organization's Active Directory(ies) into Sequence.

The ADSS copies the structure from the Active Directory based on the filter(s) that you define. Any parent container will be placed on the top (root level), i.e. Entire Organization.

The ADSS is an optional component which is installed by default during a Sequence installation.

Azure and On-Premises Support

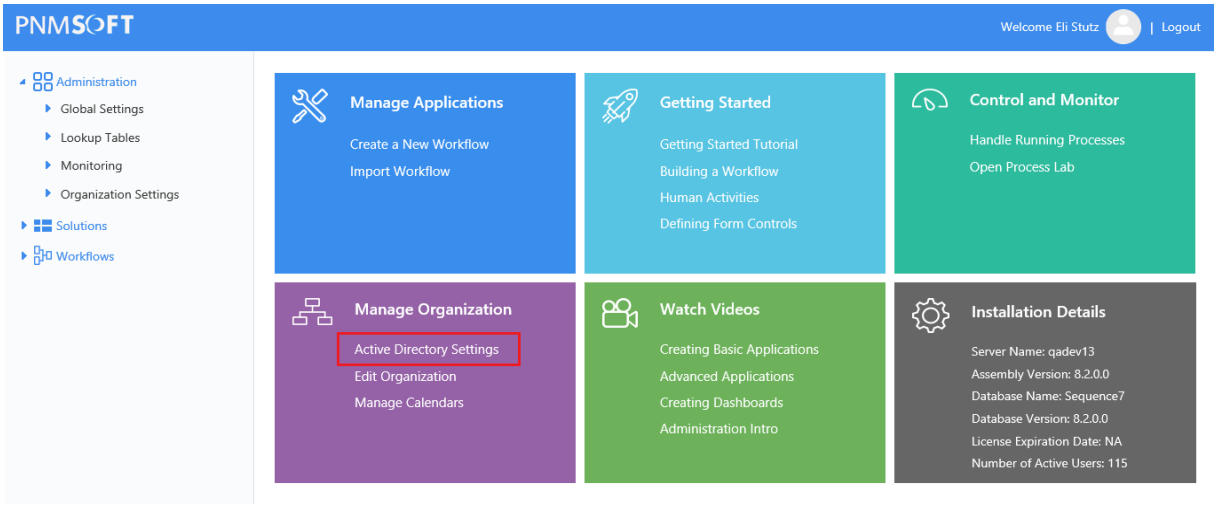
From v8.2 and above, ADSS supports secure LDAP for Active Directory on-premises, and Active Directory on Azure. Both options are available in the configuration wizard (see Step 1: The Active Directory Wizard below).

Synchronization Requirements and Planning

- Make sure you manage Sequence users in one place – in Active Directory or through Sequence.
- Make sure you manage groups in one place – in Active Directory or through Sequence
- Adding a user or group to Sequence manually and synchronizing with your Active Directory may result in an error if the user/group/OU is also captured in the filters you define in ADSS. Please refer to our troubleshooting document regarding this scenario for user or group duplicates.
- Every user object in your Active Directory that needs to be synchronized, must have at least a First Name and a Last Name. These are mandatory fields and Sequence will not synchronize users that have those fields empty.
- If you are synchronizing a whole domain, exclude 'Domain Controllers' using a filter, as it exists both as a built-in OU and built-in group. Not excluding 'Domain Controllers' will cause a conflict on the groups table in Sequence and the synchronization will fail.
- If you are synchronizing multiple domains, make sure OUs/Groups that are synchronized from the root of your LDAP path do not exist in more than one domain, as this will cause the synchronization to fail.

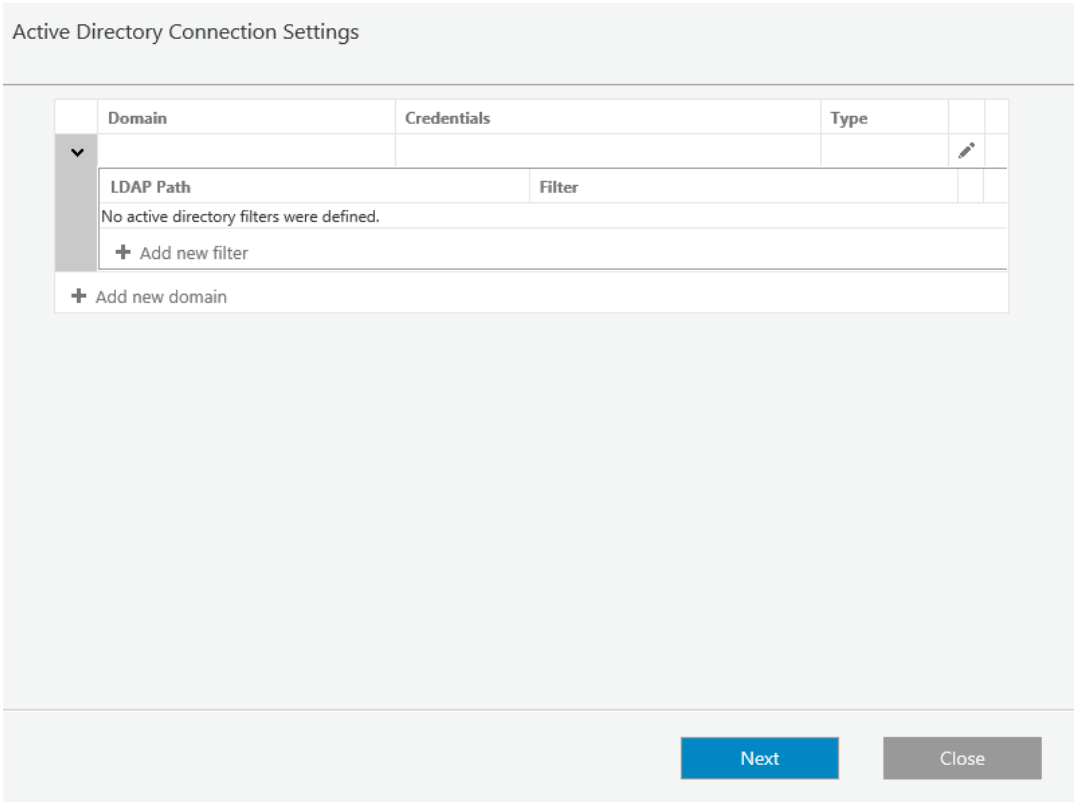
Step 1: The Active Directory Wizard

1. In the Sequence Administration site, click **Active Directory Settings**:



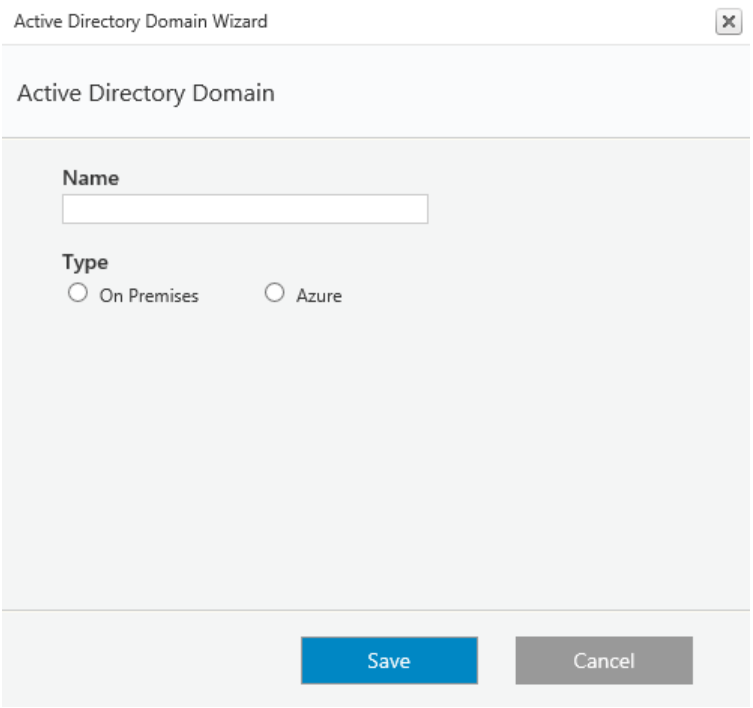
Select Active Directory Wizard

The *Active Directory Connection Settings* page opens.



Active Directory Connection Settings

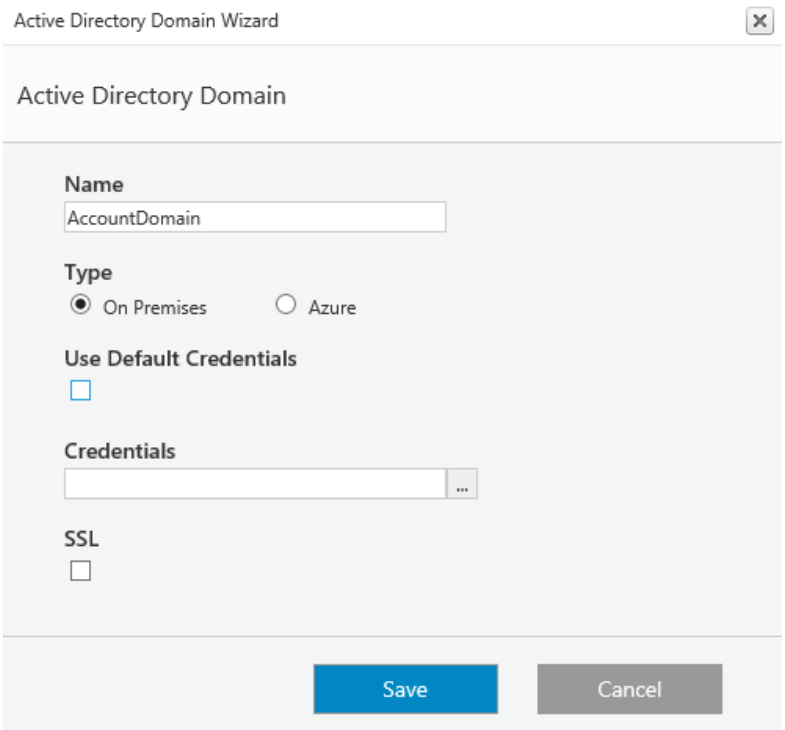
2. Click **Add new domain**. The *Active Directory Domain* screen appears.



Active Directory Domain

3. Enter a **Name** for the Domain (short name, not FQDN).
4. Select the **Type** of Active Directory: **On-Premises** or **Azure** cloud. (from v8.2 and above. Previous versions support On-Premises only).

   If you select On-Premises, enter the Credentials:

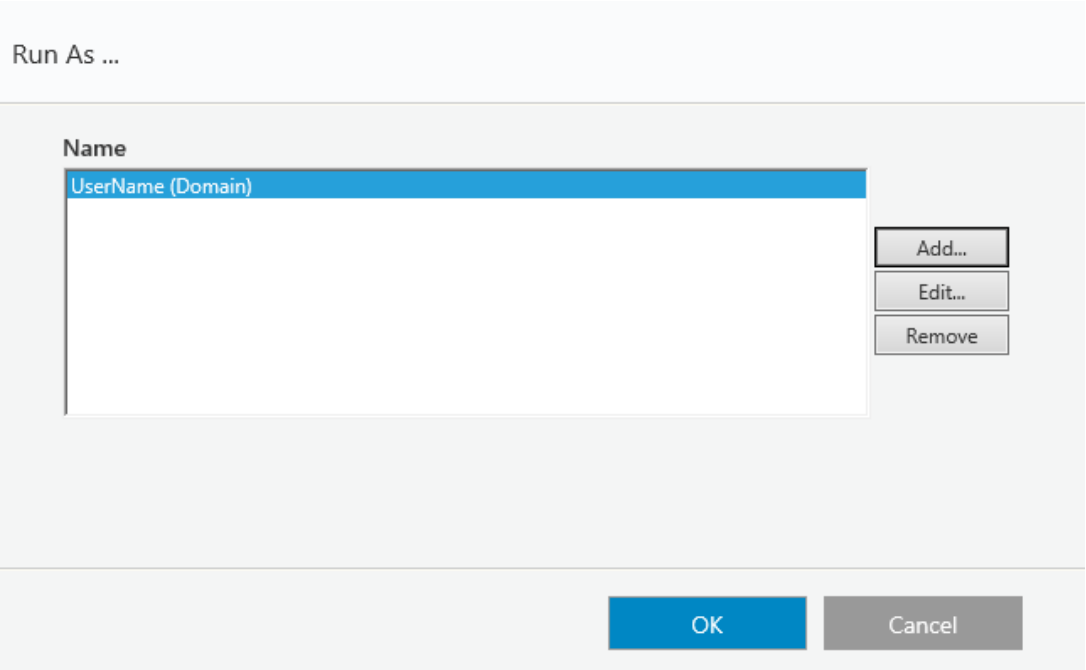


On-Premises Settings

To define specific credentials, deselect the **Use Default Credentials** checkbox and click ....

The credentials are going to be used to query the Active Directory (make sure the user has permissions to do this). If **Use Default Credentials** is checked, then the user running the Active Directory Synchronization Service will be used.

The following screen appears:

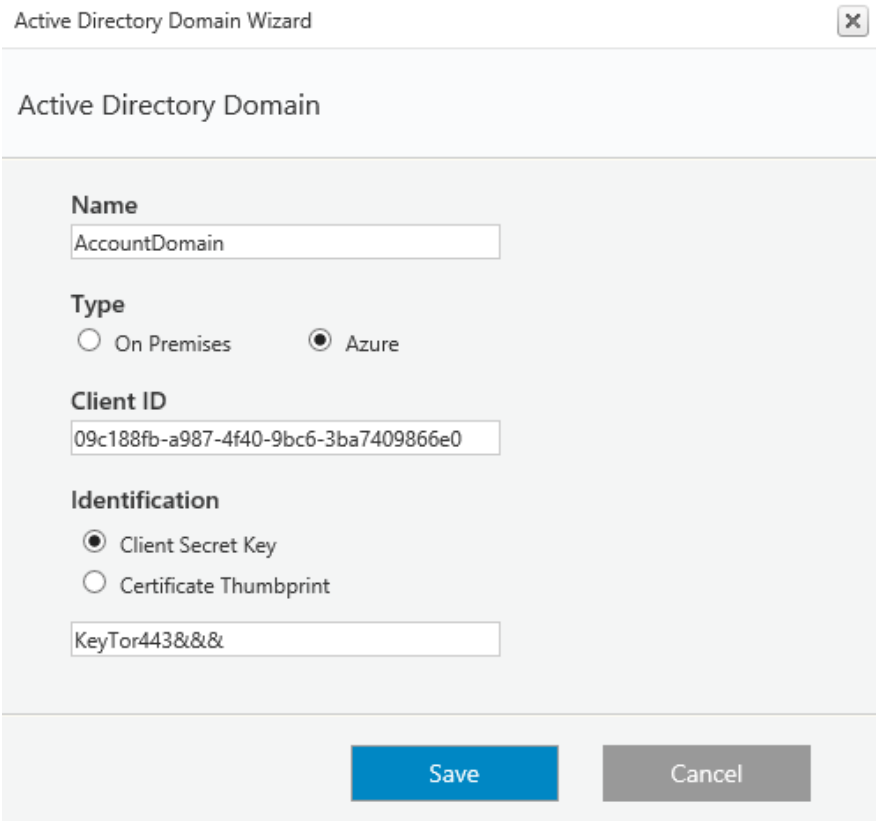


Run As

Here you can **Add**, **Edit** or **Remove** credentials. Click **OK**.

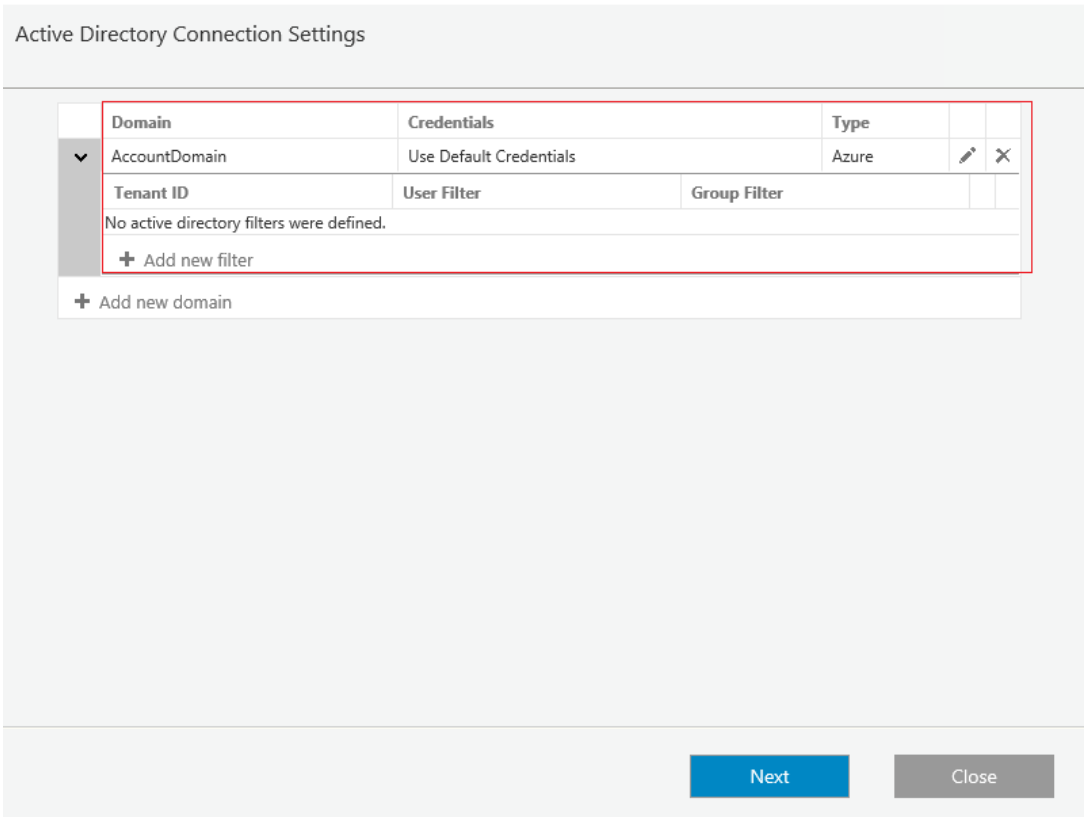
Back in the *Active Directory Domain* screen, select the **SSL** checkbox, if you wish to use SSL encryption for secure LDAP.

If you select the Type: Azure, enter the Azure **Client ID** (it must be a valid Guid), select the **Identification** type, and enter the key/certificate:



Azure Settings

Click **Save**. The new domain appears in the list of domains.



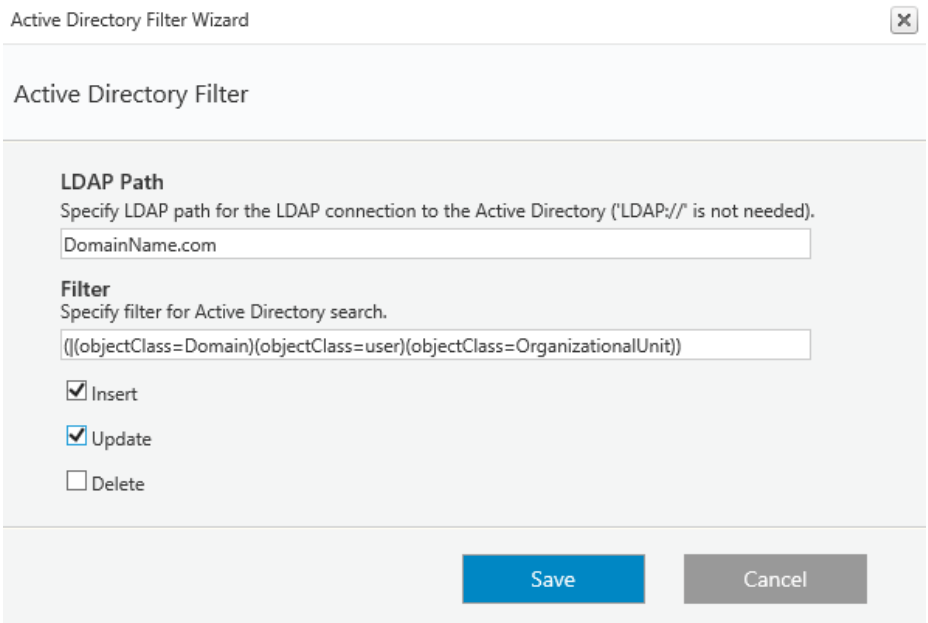
Domain Added

You can add additional domains, as necessary, using the steps above. See the section 'Multiple Domains Synchronization Considerations' below, before adding additional domains.

5. **Add Filter(s):** You can add filter(s) to domains you have added. Click **Add New Filter**.

The *Active Directory Filter* screen appears.

For On-Premises domains:



Active Directory Filter (for On-Premises Domains)

Define the following fields:

Field Name	Meaning/Content
LDAP Path	Specify LDAP path for the LDAP connection to the Active Directory ('LDAP://' is not needed).
Filter	Specify filter for Active Directory search. The LDAP syntax query that returns the hierarchical collection of groups and/or OUs and/or Containers/Folders and users to import. The default filter is set for all users and all organizational units. For further details on LDAP syntax see "*Appendix C:* Basic LDAP Syntax". The default filter imports the doman user and the organizational unit: (|(objectClass=Domain)(objectClass=user)(objectClass=OrganizationalUnit)) To add a group, use the following filter: (|(objectClass=Domain)(objectClass=user)(objectClass=OrganizationalUnit)(objectClass=group))
Insert, Update and Delete Checkboxes	Indicate the level of modifications you want to allow the Active Directory service to make. Synchronization is one way (AD to Sequence), no changes will be made to your Active Directory.
Enable Insert	If enabled, users from the Active Directory will be inserted into Sequence.
Enable Update	If enabled, users that are modified in the Active Directory will also be updated in Sequence.
Enable Delete	If enabled, users that are deleted or made inactive on the Active Directory will be made inactive in Sequence. Users are not deleted from Sequence.

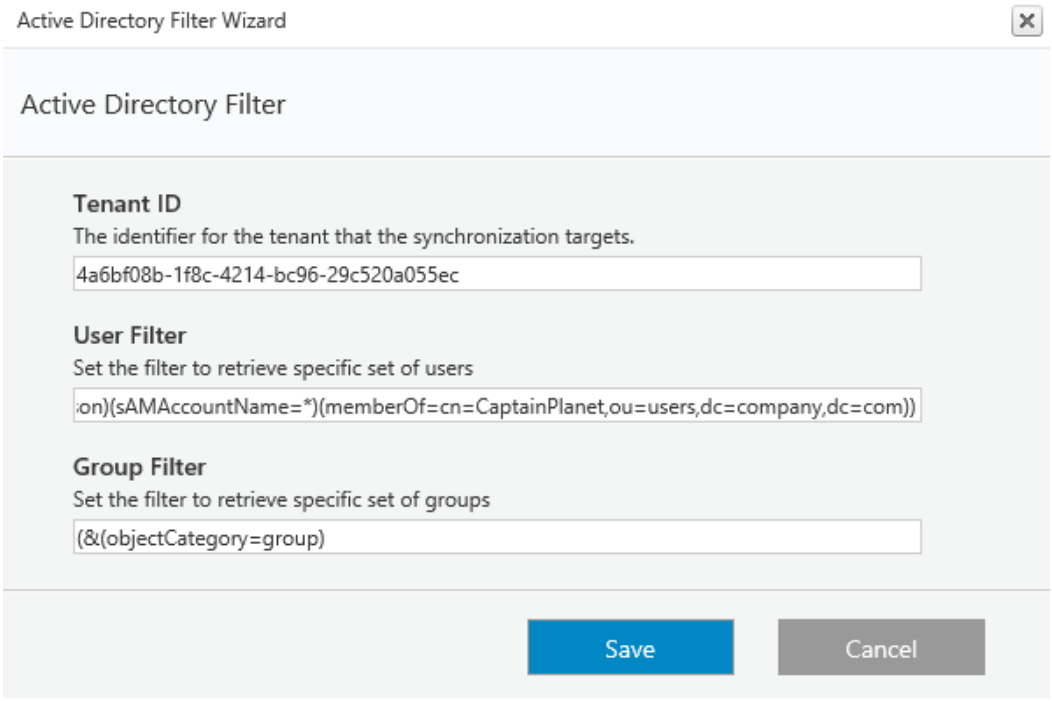
Note: Active Directory Bulk Synchronization

In Sequence v7.10 and above, the synchronization of the ADSS has been improved, and will perform synchronization from your Active Directory to Sequence quicker. This also means that by default, Active Directory data is synchronized in bulk to Sequence, overwriting changes to the organization made in Sequence. This is true even if Insert/Update/Delete options have been deselected in the screen above.

If you wish to revert back to the synchronization behavior of previous versions, which took the above settings into account (e.g. if Update is deselected, the synchronization will not overwrite records edited in Sequence), you can do so by editing the ADSS web.config file and setting this field to “false”:

`useBulkSynchronization="false"` ("true" is the default)

For Azure domains (v8.2 and above):

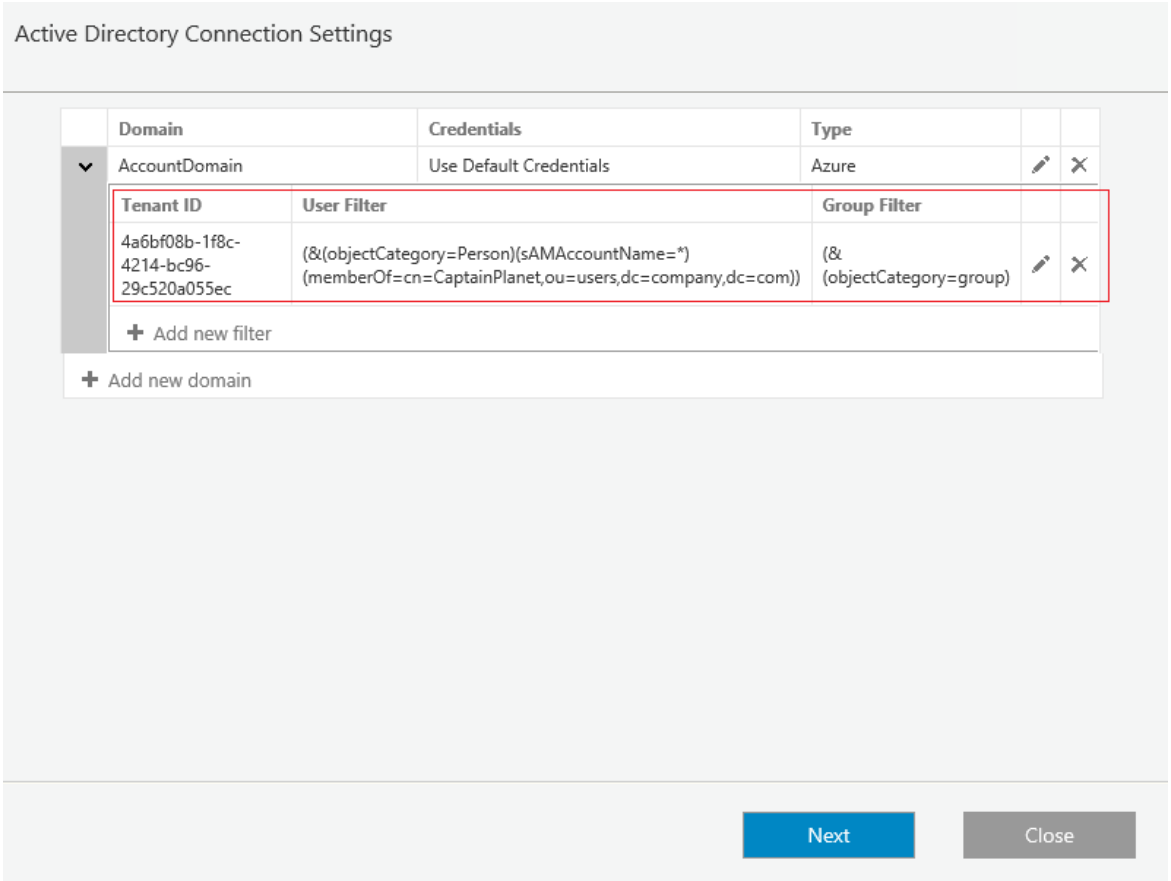


Active Directory Filter (for Azure domains)

Define the following fields:

Field Name	Meaning/Content
Tenant ID	The identifier for the Azure tenant that the synchronization targets.
User Filter	Set a filter to retrieve a specific set of users. **Note:** You must use the Azure Active Directory filter convention.
Group Filter	Set a filter to retrieve a specific set of groups. **Note:** You must use the Azure Active Directory filter convention.

6. Click **Save**. You can add additional Filters as necessary, using the steps above.

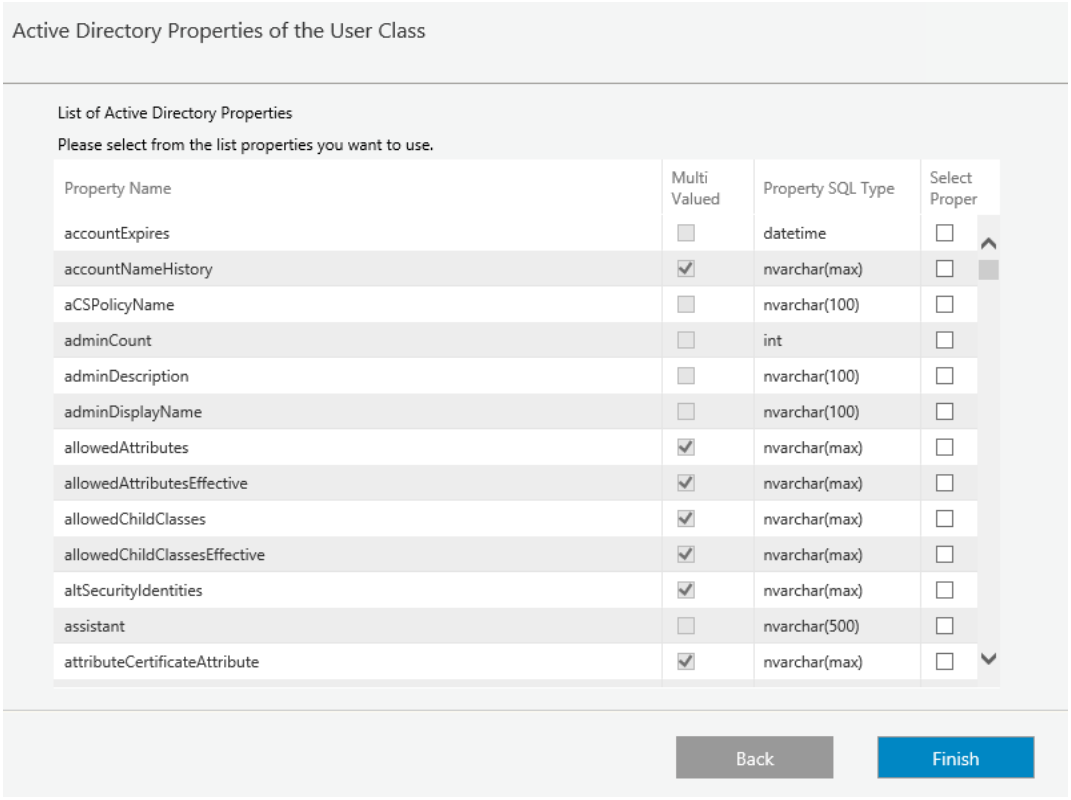


Domain and Filter Added

Click ✎ to edit existing Domains and Filters.

Click ✕ to delete existing Domains and Filters.

7. Click **Next**. The *Active Directory Properties of the User Class* page opens. The table presents a pre-selected, default set of standard user parameters.



The Active Directory Properties of the User Class Table

8. Select the attributes (in addition to the standard ones), that you wish to import from the Active Directory to Sequence and then click **Finish**.

   Note: You can import user photos from the Active Directory using the photo property. Any photos that are saved in this property will be imported into Sequence.

Any additional properties you choose to synchronize will be added to the employees table in the Sequence Database (tblEmployees). Apart from the photo, these properties will only be available on the Database table.

Multiple Domains Synchronization Considerations

Sequence supports multiple domains synchronization (version 7.7.1 and above).

Please note the following considerations for multiple domains synchronization:

1. Synchronizing a user who has the same display name on multiple domains: the user will by synchronized properly and each entry will be unique, since it will have a unique DN, meaning that a synchronized user will authenticate using his DOMAIN\username. However, the employee picker in the App Studio which is used in different activities and modules is based on employee display name, so the same display name may appear more than once.
2. When synchronizing an OU structure from multiple domains, ensure the parent name of an OU from different domains is different, for example:

Domain 1:

-> OU=Retailers-1 (Parent)

-> OU=Retailer 1

-> OU=Users

-> OU=Retailer 2

-> OU=Users

-> OU=Retailer 3

-> OU=Users

Domain 2:

-> OU=Retailers-2 (Parent)

-> OU=Retailer 1

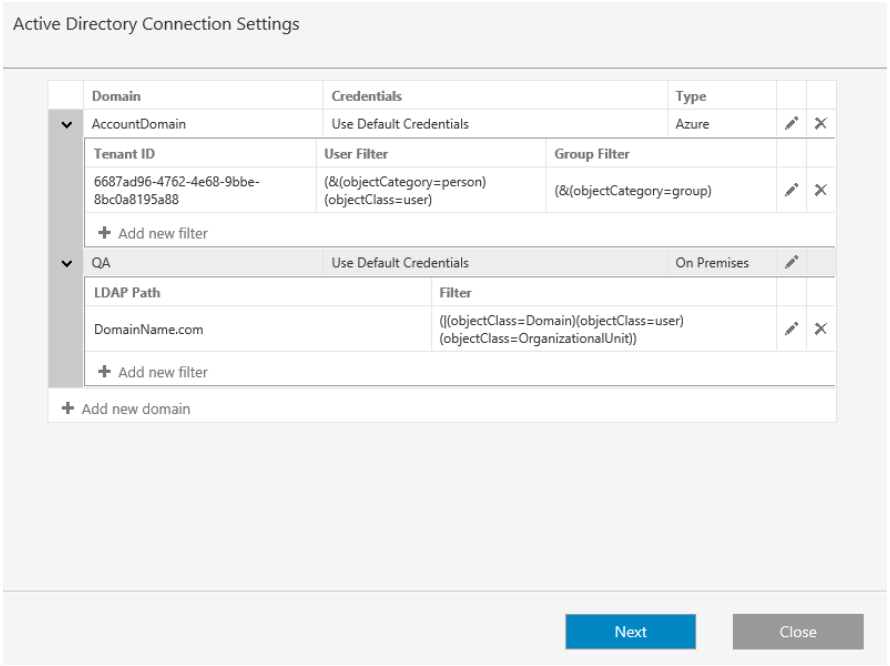
-> OU=Users

-> OU=Retailer 2

-> OU=Users

-> OU=Retailer 3

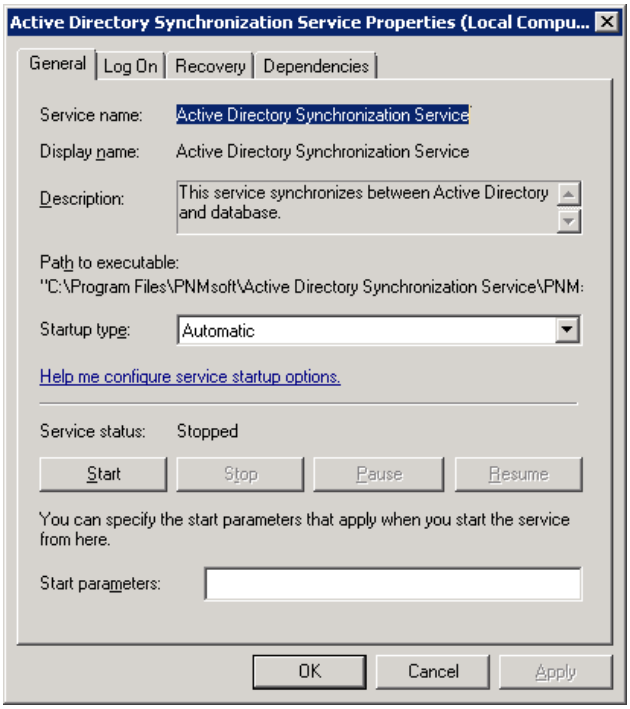
-> OU=Users



Multiple Domains

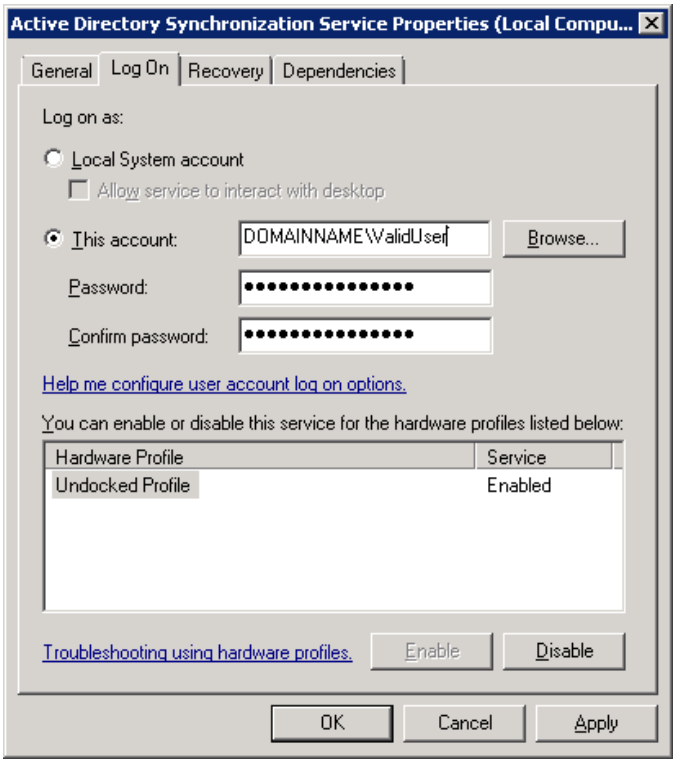
Step 2: The Active Directory Synchronization Service

1. Logon to the server where the Active Directory Synchronization Service is installed and open the Services console.
2. Locate the Active Directory Synchronization Service and open its *Properties* page. Select the *General* tab. The service should be set to *Automatic* in the **Startup Type** field.



Active Directory Synchronization Service Properties Page - General Tab

3. Select the *Log On* tab. The user configured in this screen is the one that was specified during the Sequence installation. This is the user who will query the Active Directory for the default domain and the domains that are set to use the default credentials (see step 1 point 2):



Active Directory Synchronization Service Properties Page - Log On Tab

Note: You can install the Active Directory service on multiple machines for redundancy. The Sequence engine automatically configures failover mode and no further actions are required.

Appendix A: The Active Directory Service Config File

1. Open the *Active Directory Service Config* file, located by default in `C:\Program Files\PNMsoft\Active Directory Synchronization Service`.

   The snippet of XML below is the only part that can be edited. An explanation of each key and its valid input follows:

```
<adSyncronizationThread
    type="PNMsoft.Sequence.WindowsServices.ADSS.BuiltinThreads.ADSyncronizator,
    PNMsoft.Sequence.WindowsServices.ADSS"
    interval="60"
    domain="MYDOMAIN"
    computerNameAD="MYDOMAIN"
    pageSize="100"
    serverPageTimeLimit="60"
    adUniqueKey="objectGuid"
    startSynchLoopsAt=""
    stopSynchLoopsAt=""
    useTombstonesToDetectDeletedObjects="true" />
```

Key Name	Explanation	Valid Setting	Recommended Setting
interval	the period of time in minutes that the service will pause after each update from the AD server.	Any positive integer.	720
domain	the name of the domain managed by the AD server. ***Note: the value is taken from the config file when the domain name of the filter is empty.***	any string, configured by the installer to what was entered in the Active Directory Service Settings screen	The short name of the default domain to synchronize
computerNameAD	the name of the AD server. ***Note: This key is obsolete as of Sequence version 7.4 and above.***	any string, must be equal to the domain. Configured by the installer to what was entered in the Active Directory Service Settings screen	Must be equal to the domain.
pageSize	the number of records copied from the AD server each time. Should be lowered when the AD server responds slowly to avoid timeouts.	20-500	100

Key Name	Explanation	Valid Setting	Recommended Setting
serverPageTimeLimit	the period of time in seconds that the service waits for a reply before timing out. Should be raised when the AD server responds slowly.	60-500	60
adUniqueKey	determines the unique key setting for the AD service. Valid settings are "objectGuid" to denote using the key from the AD server or "domainUserName" to denote a combination of the user's domain and NT username.	"objectGuid" or "domainUserName"	objectGuid
startSynchLoopsAt	used in combination with the stopSynchLoopsAt attribute to limit the AD queries by hour of the day.	24 hour time format - HH:mm (for example 23:34). Other formats will not be processed.	
stopSynchLoopsAt	see startSynchLoopsAt for details and limitations.	24 hour time format - HH:mm (for example 23:34). Other formats will not be processed.	
useTombstonesToDetectDeletedOjbects	If set to “true”, Sequence recognizes tombstones, which represent objects that have been deleted from the Active Directory and deletes (if group/OU) or made inactive (if user).	“True” or “False”	True
debugState	Obsolete	"On" or "Off"	

Note: Any manual change to the configuration will be overwritten in upgrades, so please record any changes you make so that you can reconfigure after upgrades.

Appendix B: Additional Options

Adding Custom Code

You can add custom code before and/or after each synchronization cycle, as follows:

1. Create a new class library project.
2. Ensure that your class inherits from `IBeforeSynchronizationCycleStarted` for code to be executed before synchronization or from `IAfterSynchronizationCycleCompleted` for code to be executed after synchronization.
3. Implement the method `OnBeforeSynchronizationCycleStarted` for code to be executed before synchronization or `OnAfterSynchronizationCycleCompleted` for code to be executed after synchronization.
4. Sign your assembly and place it on the server GAC.
5. Register the assembly you have just created in the following section of the config file, according to this sample:

```
<serviceEvents>
    <!-- Example how to add custom event handler assembly to the service
                                <eventAssemblies>
                                                    <add
   type="CustomEventAssemblyNamespace.CustomEventClass, CustomEventAssemblyName,
   Version=1.0.0.0, Culture=neutral, PublicKeyToken=ca08babd3815c17a"
   eventName="EvenetMethodName" />
                                </eventAssemblies>
                                -->
  </serviceEvents>
```

Note: Replace `EvenetMethodName` with `OnBeforeSynchronizationCycleStarted` or `OnAfterSynchronizationCycleCompleted`

Appendix C: Basic LDAP Syntax

Argument	Name	EXPLANATION & EXAMPLE
=	Equal to	This LDAP argument means a certain attribute must be equal to a certain value to be true. For example, if you want to find all objects that have the first name of John, you would use: (givenName=John) This would return all objects that have the first name of John. Parentheses are included to emphasize the beginning and end of the LDAP statement.
&	Logical AND	Use this syntax when you have more than one condition, and you want all conditions in the series to be true. For example, if you want to find all of the people that have the first name of John and live in Dallas, you would use: (&(givenName=John)(physicalDeliveryOfficeName=Dallas)) Notice that each argument is in its own set of parentheses. The entire LDAP statement must be encompassed in a main set of parentheses. The & operator means that each argument must be true for this filter to apply to your object.
!	Logical NOT	This operator is used to exclude objects that have a certain attribute. Suppose you need to find all objects except those that have the first name of John. You would use the following statement: (!givenName=John) This statement would find all objects that do not have the first name of John. Notice that the ! operator goes directly in front of the argument and inside the argument's set of parentheses. Because there is only one argument in this statement, it is surrounded with parentheses for illustration.
*	Wildcard	Use the wildcard operator to represent a value that could be equal to anything. One such situation might be if you wanted to find all objects that have a value for title. You would then use: (title=*) This would return all objects that have the title attribute populated with a value. Another example might be if you know an object's first name starts with Jo. Then, you could use the following to find those: (givenName=Jo*) This would apply to all objects whose first name starts with Jo.

For more information please review the following articles:

- LDAP Query Basics - http://technet.microsoft.com/en-us/library/aa996205(v=exchg.65).aspx
- Search Filter Syntax - http://msdn.microsoft.com/en-us/library/aa746475(v=vs.85).aspx
- Active Directory: LDAP Syntax Filters - http://social.technet.microsoft.com/wiki/contents/articles/5392.active-directory-ldap-syntax-filters.aspx