LDAP Authentication

About LDAP Support

You can delegate Controller UI authentication and authorization to external directory servers that comply with LDAP (Lightweight Directory Access Protocol) version 3.

While the Controller should be able to work with any LDAPv3-compliant server, the Controller has been verified to work with these LDAP products:

- Microsoft Active Directory for Windows Server 2008 SP2+
- OpenLDAP, 2.4+

To configure LDAP authentication in the Controller, you need to configure connection settings to the LDAP server and the queries that return user or group data. By mapping LDAP groups to roles, you can provision permissions in the AppDynamics Controller based on LDAP groups.

What Happens if the LDAP Server Becomes Unavailable?

If the LDAP server configured for Controller authentication becomes unavailable for any reason, the Controller falls back to local user authentication. Given this possibility, you should provision local user accounts in AppDynamics for the administrative users who will need access if the LDAP server becomes unavailable.

What Happens if a User Cannot be Found in the LDAP Directory?

If a user cannot be found in the LDAP directory, the authentication failure event is logged as a warning. The user, whether it is a regular controller user or a REST client user, may still be authenticated through local authentication.

Preparing the LDAP Directory for AppDynamics Integration

To use an LDAP authentication provider, your AppDynamics Controller needs to be able to connect to the external LDAP server. A good practice is to create a user account in LDAP specifically for the Controller to use to authenticate itself to the server and run the
queries. The Controller user only needs to have search privileges in LDAP.

While you can map existing LDAP group definitions to roles in AppDynamics, your existing groups may not correspond directly to roles in AppDynamics. The easiest way to map LDAP groups to Controller roles is to create a group in LDAP for each role you want to be mapped in AppDynamics. LDAP groups for each role provides you with a manageable, 1-to-1 correspondence between your LDAP groups and AppDynamics roles.

For example, a possible LDAP group scheme for mapping in AppDynamics would be:

- AppDynamics-AppA-ReadOnly
- AppDynamics-AppA-Admins
- AppDynamics-AppA-DashboardViewers
- AppDynamics-AppB-ReadOnly
- AppDynamics-AppB-Admins
- AppDynamics-AppB-DashboardViewers

The sample group names imply having custom roles in AppDynamics targeted to specific applications, AppA and AppB.

Naming the groups with a common prefix, as the AppDynamics- prefix in our sample, allows you to use a relatively simple LDAP group filter. A group filter for the sample groups could be:

```
(&(objectClass=group)(cn=AppDynamics-*))
```

**Using Paged Results for Large Result Sets**

LDAP servers are sometimes configured to limit the number of entries it can return in a query response. If the results of your user or group query exceed that limit, AppDynamics reports a max_results_exceeded error.

To avoid this error, first try to refine your query filter to produce a smaller result set. The results must include the users who will need to access the AppDynamics UI.

If your LDAP server supports it, you can also enable paged results in the Controller LDAP configuration. With paged results, the LDAP server divides the result set into separately transmitted blocks.

The paged results feature applies to the behind-the-scenes interaction between the AppDynamics Controller and the backend LDAP server. It does not affect the UI view of the data.

**LDAP Authentication with a SaaS AppDynamics Controller**

Depending on your organization’s security policies, it may not be possible to use LDAP authentication with the SaaS AppDynamics Controller, since doing so requires opening your firewall to permit Controller access to your corporate LDAP server.

However, if you do want to enable LDAP authentication with SaaS AppDynamics Controller, you will need to permit access through the firewall for the IP range of 69.27.44.0/24, the IP address range assigned to AppDynamics SaaS Controllers. The firewall rule should allow incoming LDAP requests from the Controller at the LDAP port you configure.

**Before Starting**

To perform LDAP configuration, you must have:

- An LDAP server. There is a one-to-one correspondence between an AppDynamics account and an LDAP server.
• An account on an AppDynamics SaaS or on-premises Controller
• Account administrator privileges on the AppDynamics Controller, as described in User Management.
• Network connectivity between your LDAP server and the Controller. If using a SaaS Controller, the LDAP server may not be accessible to the Controller without enabling access through your network firewall. See LDAP Authentication with a SaaS AppDynamics Controller.

Configuring LDAP Authentication

At a high level, the steps for setting up LDAP authentication include:

• Configure the connection to the LDAP server.
• Configure and test the LDAP query that returns users to be provisioned in the AppDynamics Controller.
• Configure the LDAP query that returns the LDAP groups to be mapped to AppDynamics roles.
• Map the users or groups to roles in AppDynamics.

Configure the Connection to the LDAP Server

As an administrator or account owner in the Controller UI, you can configure LDAP authentication from the Authentication Provider tab under Settings > Administration.

If the user or group query that you need to use will return more entries than permitted by the LDAP server and the server support paged results, configure paged results as follows:

• Enable Paging: Check this option to have the Controller request paged results from the server when submitting user or group queries.
• Page Size: Enter the number of entries per round-trip from the AppDynamics Controller to the LDAP server. The default is 500.

The page size should be the total number of entries to be returned divided by the number of round trips between the LDAP server and the Controller that are tolerable. For example, if you expect to receive 1200 results in a query and you can tolerate a maximum of two round trips, set the page size to 600 (1200 /2). See Using paged results for large result sets for more information.

Configure the LDAP connection settings:

• Host: Required address of the LDAP server.
• Port: Port on which the LDAP server listens. The default is 636 for an SSL connection and 389 if not using SSL. Required.
• Use SSL: Enabled by default to use a secure connection to the LDAP server. Clear if not using SSL.
• Enable Referrals: Enabled by default to support LDAP referrals. A referral is when an LDAP server forwards an LDAP client request to another LDAP server. Each referral event is referred to as a hop.
• Maximum Referral Hops: The maximum number of referrals that AppDynamics will follow in a sequence of referrals. The default is five.
• Bind DN: Distinguished Name of the user on the LDAP Server on whose behalf the AppDynamics application searches. Required.
• Password: Password of the user on the LDAP server. Required. Your settings should look something like this:
Configure Users

In the LDAP configuration page, configure information to find LDAP users:

- **Base DN**: Location in the LDAP tree to begin recursively searching for users. Required.
- **Filter**: Optional LDAP search string that filters the items matched from the base DN. See RFC2254 for information about LDAP search filters.
- **Login Attribute**: The LDAP field that corresponds to the username users will enter when logging in to the AppDynamics UI. The default is `uid`. For Active Directory, this would typically be `sAMAccountName`.
- **Display Name Attribute**: The LDAP field to use as the user’s display name.
- **Group Membership Attribute**: Optional user group membership field. Recommended for faster retrieval.
- **Email Attribute**: Optional user email address.

---

**Configure Users**

In the LDAP configuration page, configure information to find LDAP users:

- **Base DN**: Location in the LDAP tree to begin recursively searching for users. Required.
- **Filter**: Optional LDAP search string that filters the items matched from the base DN. See RFC2254 for information about LDAP search filters.
- **Login Attribute**: The LDAP field that corresponds to the username users will enter when logging in to the AppDynamics UI. The default is `uid`. For Active Directory, this would typically be `sAMAccountName`.
- **Display Name Attribute**: The LDAP field to use as the user’s display name.
- **Group Membership Attribute**: Optional user group membership field. Recommended for faster retrieval.
- **Email Attribute**: Optional user email address.
The Test Query button checks the connection. If successful, a screen displays the first few users returned by the query. (The test does not return the entire result set if the result set is large.)

Configure Groups

Optionally, you can map LDAP groups to user roles in the AppDynamics Controller. To do this, you need to set up the LDAP query that returns the LDAP groups to map, as follows.

- **Base DN:** Location in the LDAP tree to begin recursively searching for groups. Required.
- **Enable Nested Groups:** Option to include nested LDAP groups to a depth of 10.
- **Filter:** Optional LDAP search string that filters the items matched from the base DN. See RFC2254 for information about LDAP search filters.
- **Name Attribute:** The LDAP field that contains the name of the group. Default is `cn`. Required.
- **Description Attribute:** The LDAP field that contains a description of the group. Optional.
- **User Membership Attribute:** Identifies members of the groups. Optional.
- **Referenced User Attribute:** Optional child attribute of the User Membership Attribute. Disabled if the parent is empty. Identifies property of the user that the user membership attribute contains.
The Test Query button checks the connection. If successful, the first few groups returned by the query are shown. You can now assign permissions in the AppDynamics Controller to users or groups.

**Assign AppDynamics Permissions to an LDAP User**

1. In the Security Configuration window, click the **Users** tab.
   If LDAP is enabled and correctly configured, the AppDynamics Controller fetches the user names from the LDAP server.
2. Select the name of the user to whom you want to assign permissions.
3. In the Roles panel, check the roles that you want to assign to this user. You can assign multiple roles to a user.
Assign AppDynamics Permissions to an LDAP Group

LDAP Group configuration is optional.

1. In the Security Configuration window, click the Groups tab.
2. If LDAP is enabled and correctly configured, AppDynamics fetches the group names in LDAP.
3. Select the name of the group to which you want to assign permissions.
4. In the Roles panel, check the roles that you want to assign to this group. You can assign multiple roles to a group.
5. Click Save.

Configuring the LDAP Cache Synchronization Frequency

The Controller keeps information about LDAP users and groups in a local cache. It regularly connects to the LDAP server to synchronize its cache with the LDAP server.

The Controller caches information about users and group membership. It does not cache user passwords. Accordingly, the Controller authenticates the user credentials against the LDAP server at the start of every user session.

If a user account is removed from LDAP, the change is reflected immediately; that is, the user will not be able to log in to the Controller UI from that point. However, if the user has an existing session in the Controller UI, that session continues until the user logs out or the session expires.

If the user’s access to the Controller is based on group membership and the user is removed from the group but maintains an account
in the LDAP server, the user will be able to log in to the Controller until the next time synchronization with the LDAP server occurs. By the default synchronization frequency setting, this ability to access the Controller UI could continue for up to an hour.

You can modify the default synchronization frequency of one hour as described in the following procedures.

**Configure the LDAP Synchronization Frequency**

1. Stop the Controller application server:
   - **On Linux**, run:
     ```
     platform-admin.sh stop-controller-appserver
     ```
   - **On Windows**, run this command from an elevated command prompt, which you can open by right-clicking the Command Prompt icon in the Windows Start menu and choosing **Run as administrator**:
     ```
     platform-admin.exe cli stop-controller-appserver
     ```

2. **Open the** `<Controller-Installation-Directory>/appserver/glassfish/domains/domain1/config/domain.xml` **file for editing.**
3. **In the** `<jvm-options>` **element, add a system property named** `appdynamics.ldap.sync.frequency` **with the desired synchronization frequency in milliseconds.**
   For example, to have the Controller synchronize to the LDAP server every 15 minutes (900000 milliseconds), add:
   ```
   <jvm-options>-Dappdynamics.ldap.sync.frequency=900000</jvm-options>
   ```
   The default is 3600000 milliseconds (1 hour).
4. **Save the file.**
5. **Restart the Controller app server:**
   - **On Linux**, run:
     ```
     platform-admin.sh start-controller-appserver
     ```
   - **On Windows**, run the following in an elevated command prompt:
     ```
     platform-admin.exe cli start-controller-appserver
     ```