Technical Document

NiagaraAX LDAP / Active Directory Configuration Guide

AX-3.8 and AX-3.7u1

November 6, 2013

Niagara^{AX} LDAP / Active Directory Configuration Guide

Copyright © 2013 Tridium, Inc.

All rights reserved.

3951 Westerre Pkwy., Suite 350

Richmond

Virginia

23233

U.S.A.

Confidentiality Notice

The information contained in this document is confidential information of Tridium, Inc., a Delaware corporation ("Tridium"). Such information, and the software described herein, is furnished under a license agreement and may be used only in accordance with that agreement.

The information contained in this document is provided solely for use by Tridium employees, licensees, and system owners; and, except as permitted under the below copyright notice, is not to be released to, or reproduced for, anyone else.

While every effort has been made to assure the accuracy of this document, Tridium is not responsible for damages of any kind, including without limitation consequential damages, arising from the application of the information contained herein. Information and specifications published here are current as of the date of this publication and are subject to change without notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia.

Trademark Notice

BACnet and ASHRAE are registered trademarks of American Society of Heating, Refrigerating and Air-Conditioning Engineers. Active Directory, Microsoft, Excel, Internet Explorer, Windows, Windows Vista, Windows Server, and SQL Server are registered trademarks of Microsoft Corporation. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Mozilla and Firefox are trademarks of the Mozilla Foundation. Echelon, LON, LonMark, LonTalk, and LonWorks are registered trademarks of Echelon Corporation. Google Chrome is a trademark of Google Inc. OpenLDAP is a registered trademark of the OpenLDAP Foundation. Tridium, JACE, Niagara Framework, Niagara^{AX} Framework, and Sedona Framework are registered trademarks, and Workbench, WorkPlace^{AX}, and ^{AX}Supervisor, are trademarks of Tridium Inc. All other product names and services mentioned in this publication that is known to be trademarks, registered trademarks, or service marks are the property of their respective owners.

Copyright and Patent Notice

This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution of those products, subject to the contracts that authorize such distribution. It may not otherwise, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc.

Contents

Preface

- LDAP FAQs
- Document Change Log

LDAP FAQs

The following are frequently asked questions (FAQs) about using LDAP, Active Directory, and Kerberos with NiagaraAX.

Q: Can I use SSL with LDAP?

A: Yes, in fact it is recommended to configure NiagaraAX platforms and stations for SSL. For newer JACE models and all Windows-based hosts (using the Hotspot JVM), refer to the *NiagaraAX SSL Connectivity Guide* for related details. For older JACE models (JACE-2 or JACE-4/5 series), which use the IBM J9 JVM, refer to the *NiagaraAX CryptoService (SSL)* engineering notes document. Note that Kerberos is not supported on these older JACE models, however.

Q: Can a NiagaraAX system use a combination of LDAP or Active Directory along with the "network user" feature in a NiagaraNetwork?

A: No, the NiagaraAX "network user" feature is incompatible with using LDAP or Active Directory user services (no "hybrid system" supported). All centralized user management is provided by the LDAP server or Active Directory server, and each station requires a user service sourced from the `ldap` module. However, "local" station users, unique to each station, are still supported.

Q: Is Kerberos always associated with LDAP in NiagaraAX?

A: Starting in AX-3.8, Kerberos is an available authentication method for LDAPv3 compatible user services in the ldap module (LdapV3UserService, LdapV3ADUserService). Alternatively, you can use another authentication method instead—for example DIGEST-MD5, CRAM-MD5, or simple (clear text). Currently outside of these two LDAP-based user services, Kerberos is not used in NiagaraAX.

Q: Do properties of an LDAP user service such as "Password Strength" (to configure strong passwords in AX-3.8) and various "Lockout" properties apply to users under the service?

A: Yes, but only to "local" users created in the station—and *not* to any LDAP users. The same is also true of the "Password Configuration" properties of the LDAP user service and users, first introduced in AX-3.7. These properties that define periodic password expirations and enforce "unique passwords" apply only to local station users, and *not* to any LDAP users.

Q: Can an AX-3.8 station support an older LDAPv2 level server or Active Directory using the newer LDAPv3 compatible user services (LdapV3UserService, LdapV3ADUserService)?

A: Yes, these newer LDAP user services are backwards-compatible with an LDAPv2-based system. However, Kerberos authentication is not available in this scenario.

Q: Can I configure my NiagaraAX 3.8 stations to run in FIPS mode (FIPS 140-2) and also use LDAPv3 with Kerberos authentication?

A: No, when running in FIPS mode, the set of permitted cryptographic algorithms is smaller—only algorithms that are FIPS-approved may be used. Due to these restrictions, Kerberos cannot be used when running in FIPS mode, as the algorithms it requires are not supported by the FIPS cryptographic provider.

Document Change Log

Updates to this *NiagaraAX LDAP / Active Directory Configuration Guide* are listed below.

- Publication: November 6, 2013
  Initial document, replacing previous PDF-only "LDAP User Service Guide". Content of this document is also available in the Workbench help system as "Doc Ldap AD" (module `docLdapAD`).

CHAPTER 1

LDAP / Active Directory Quick Start

This section lists requirements and provides procedures to configure one of the alternate user services found in the AX-3.8 `ldap` module or AX-3.7 `ldap` module.

System requirements

A NiagaraAX LDAP integration has the following system requirements:

- LDAP environment
- NiagaraAX platforms and licensing
- Station prerequisites
- Kerberos prerequisites

LDAP environment

To effectively use any of the user services from the `ldap` palette, the NiagaraAX host running a station must be installed on a network with an existing LDAP (or Active Directory) server, where support is available for LDAPv2, and if using AX-3.8 stations, either LDAPv3 or LDAPv2. When configuring the user service in the station, you will need information supplied by the administrator of that LDAP system.

In AX-3.8 stations (only), if using Kerberos authentication for either of the LDAPv3-compatible user services, information will also be needed from the Kerberos administrator (possibly the same person).

NiagaraAX platforms and licensing

Any NiagaraAX platform (JACE, Supervisor) running AX-3.8 or later is compatible with any of the user services in the `ldap` module, providing that the `ldap` module is installed.

If using either AX-3.8 LDAPv3-compatible user service (LdapV3UserService or LdapV3ADUserService), the NiagaraAX host platform must be licensed with the feature `“ldapv3”`. A sample license line is below.

```
<feature name="ldapv3" expiration="never" kerberos="true" parts="LDAPV3_PART"/>
```

Note: *Kerberos authentication requires the attribute* `kerberos=”true”`*, as shown above.*

Note also that Kerberos is not supported on any “J9 Java VM” (JACE-2/4/5 series) platform.

Note the LDAPv2-compatible user services (LdapUserService, ActiveDirectoryService), also available in AX-3.7, *do not* require host licensing. These are effectively the same LDAP user services provided since NiagaraAX-3.1. However, note that these user services *do not offer Kerberos* as an authentication choice.

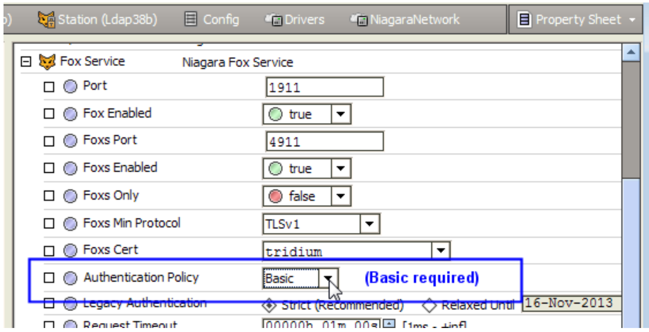
Station prerequisites

As with previous versions of NiagaraAX LDAP user services, all current versions (LDAPv3 and LDAPv2) *may not* be used with the (default) FoxService authentication scheme of **`Digest`**, nor with the (default) WebService authentication scheme of **`Cookie Digest`**.

Therefore with *any* LDAP user service:

- If you want LDAP user login via Workbench, you must set the FoxService's "Authentication Scheme" property to **`Basic`** (see Figure 1-1).

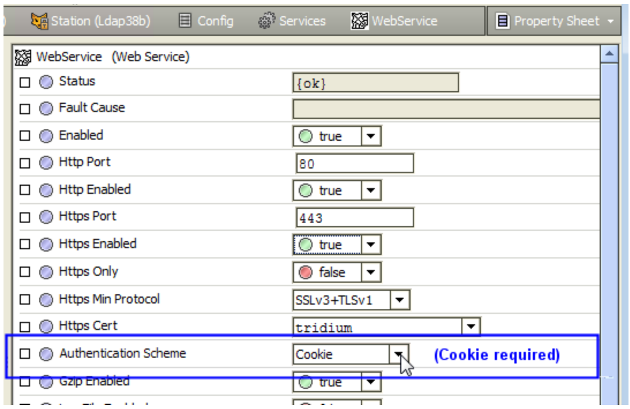
Figure 1-1 *Set Config > Drivers > NiagaraNetwork > FoxService property Authentication Scheme to Basic*



Fox Service	Niagara Fox Service
Port	1911
Fox Enabled	true
Foxs Port	4911
Foxs Enabled	true
Foxs Only	false
Foxs Min Protocol	TLSv1
Foxs Cert	tridium
Authentication Policy	Basic **(Basic required)**
Legacy Authentication	◇ Strict (Recommended) ◇ Relaxed Until 16-Nov-2013
Request Timeout	00000h 01m 00s [1ms - +inf]

- If you want LDAP user login via browsers, you must change the station's WebService property "Authentication Scheme" to **`Cookie`** (see Figure 1-2).

Figure 1-2 *Set Config > Services > WebService property Authentication Scheme to Cookie*



WebService (Web Service)	
Status	{ok}
Fault Cause	
Enabled	true
Http Port	80
Http Enabled	true
Https Port	443
Https Enabled	true
Https Only	false
Https Min Protocol	SSLv3+TLSv1
Https Cert	tridium
Authentication Scheme	Cookie **(Cookie required)**
Gzip Enabled	true

Note best security practices recommend the **`Digest`** authentication scheme. However, that is not possible when using an LDAP user service. Therefore, we strongly recommend that you turn on the SSL features in the FoxService ("Foxs Enabled") and WebService ("Https Enabled"). This will help keep your credentials secure.

Alternatively, in AX-3.8 you can use Kerberos authentication, which is secure even over "cookie" authentication. Note SSL is still recommended, just as it is when using Digest authentication.

Kerberos prerequisites

In AX-3.8, one of the new features introduced with the LdapV3ADUserService and LdapV3UserService is the ability to use Kerberos authentication for an LDAP user to log into a station. Kerberos is a widely-used authentication protocol, and helps keep your credentials and station safe. It allows for a "single sign on" (SSO) environment, such that your initial sign-on to your local machine results in a "ticket" automatically used to access other system resources, without need for further sign on (supplying credentials).

Note: *Single sign-on Kerberos access is available to a station running on a Windows-based host, but user credentials are still required when accessing any Kerberos-configured station running on a (QNX-based) "Hotspot" JACE host when using a browser. For more details, see "Browser access via Kerberos" on page 2-7.*

In addition, note that older JACE platforms using the "J9 Java VM" (JACE-2/4/5 series) do not support Kerberos authentication. You must configure their station to authenticate directly to the LDAP server or Active Directory server, using the "SimpleAuthenticator" component.

Kerberos uses a slightly different setup than other LDAPv3 features, and it is more complicated to configure.

In order to prepare for Kerberos authentication, follow this high-level process:

1. Contact your Kerberos administrator and get the following information:
   - Kerberos realm name (should be in UPPERCASE).
   - Key Distribution Center URL.
   - Ask your administrator to set up a *service name* for your station. This should be in the form: `http/somename.domain.com`
     where `domain.com` is your realm, and `somename` is the name by which you will access your station via the browser.
     This account *must be trusted for delegation* (the admin can set this up). If you are not planning for Kerberos authentication via the browser, you can use a regular username (not a service).
   - Get either a *keytab file* or else a password for the service or user obtained above. Services typically require a keytab file, whereas users typically use a password.

   Related details are in the "Configuring the Kerberos Authenticator component" on page 1-12.
2. Set up your PC for Kerberos authentication. See the "Additional Kerberos client-side setup" on page 1-13 section for more details.
   - If you plan on logging in through Workbench, set up your `krb5.conf` file to include the line `"forwardable=true"`.
   - If you are using Windows and would like to use your native Kerberos ticket, set your Windows registry `AllowTgtSessionKey` value to `1`.
   - If you will be logging in to Kerberos using a browser, set up your browser(s) for Kerberos authentication. See the "Kerberos via the browser" appendix for more details on configuring different browsers.
3. Set up your station to use Kerberos.
   - Note separately mentioned station prerequisites. See "Station prerequisites" on page 1-2.
   - Replace the standard UserService with the LdapV3ADUserService or LdapV3UserService. See "Add and configure the LDAP user service" on page 1-3 and "Configure the LdapConfig or ActiveDirectoryConfig child" on page 1-8.
   - Configure the Kerberos Authenticator child of the LdapConfig or ActiveDirectoryConfig component. See "Configuring the Kerberos Authenticator component" on page 1-12.

Add and configure the LDAP user service

Note: *See "System requirements" on page 1-1 for prerequisites.*

To get started, perform the following tasks (do this for each station at the job):

- Add and configure the LDAP user service
- Configure any needed local users
- Configure User Prototypes

Add the LDAP service offline

Although you can try "swapping in" an LDAP user service in a running station, error messages may occur and results can be unpredictable.

Therefore, we recommend that for any existing station with the standard **`UserService`** (or an LDAP user service to be replaced by a different type), that you *save and edit that station offline* (`config.bog` file), using Workbench. In your offline edit, you swap in a different user service from the `ldap` palette.

If a remote station, that means you first use the platform **`Station Copier`** to save that station locally. After editing and saving that `config.bog` file, you use the **`Station Copier`** to install it back in that same host. The following procedure walks through these basic steps, using AX-3.8 or later Workbench.

To add an LDAP user service

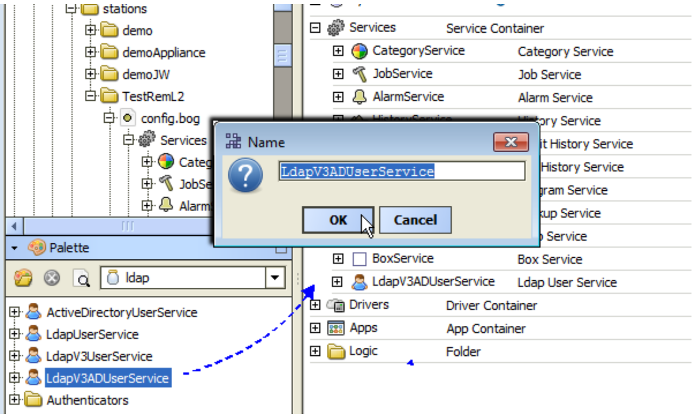
Step 1 In Workbench, open a platform connection to the host (JACE) and use the **`Station Copier`** tool to save a local copy of its station.

Note: *If working on a locally running station, such as on a Supervisor, you would (instead) open a local platform connection and* **`Stop`** *that station from the* **`Application Manager`** *view, before going to the next step.*

Step 2 In the Workbench Nav tree, expand your host's file system and navigate to that station folder, expanding the `config.bog` file to open its Config, **`Services`** container.

Step 3 Open the **`ldap`** palette in your side bar (see "Using the palette side bar" in the *User Guide* for general details).

Step 4 From the **`ldap`** palette, *drag* (or copy and paste) the needed user service into the **`Services`** container.

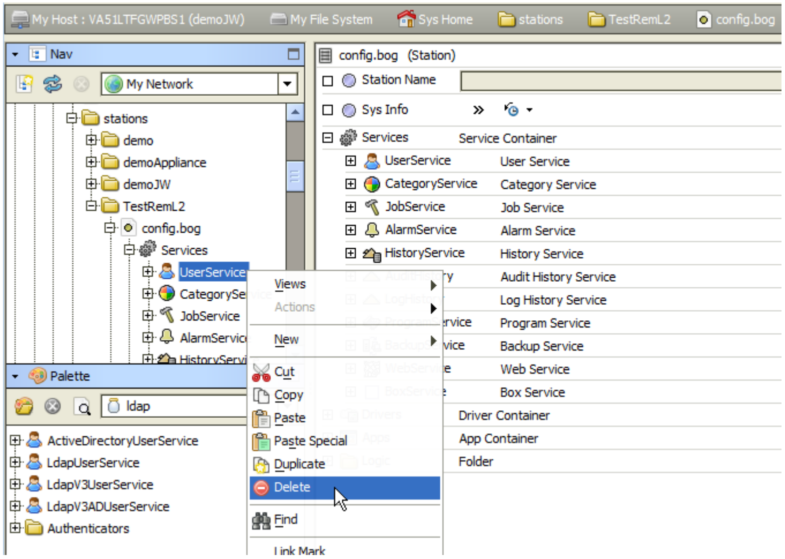


In the popup **`Name`** dialog, you can rename it—or, simply use the default name.

The new LDAP user service is now in the station's **`Services`** container.

Step 5 If the station's previous user service has local **`Users`** and/or **`User Prototypes`** that you would like to reuse, copy them at the same level under the new LDAP user service.

Step 6 In the station config.bog file, select and *delete* the existing user service (e.g. **`UserService`**).

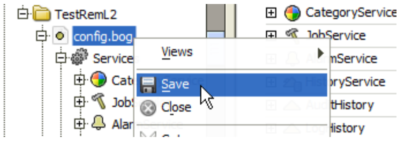


Step 7 In the new LDAP user service, assign a strong password to the built-in, local "admin" user.

- In AX-3.8, the default user password strength is 12 characters total minimum, with at least one upper case, one lower case, and one digit (numeral). This is configurable in the "Password Strength" properties of the user service.
- In AX-3.7, the minimum strong password is fixed (8 characters, at least one letter and one digit or special character).

Alternatively, you can create another "super user" account with a strong password, and then disable the (well known) "admin" user.

Step 8 Right-click the config.bog file and select 🖫 **`Save`**.



You can continue to work offline to do more configuration, or else reinstall the station (if a JACE) or else restart the local station (if a Supervisor), and work on the station while it is running. This latter online method is what is documented here.

Step 9 Open a platform connection to the host (JACE) and use the **`Station Copier`** tool to install the modified station back to the JACE.

Note: *If working on a locally running station, such as on a Supervisor, you would (instead) open a local platform connection and* **`Start`** *that station from the* **`Application Manager`** *view.*

Allow sufficient time for the station to restart. If a JACE, note that a station transfer results in a controller reboot first.

Step 10 In Workbench, open a station connection to the host as the admin (or if created, other super user) user.

⚠ ***Caution*** *By default (unless you already changed this with the station opened offline), the "admin" user in any of the LDAP user services has a "blank" password—something you should definitely change immediately!*

In most cases it is recommended you create another local user that is a super user, assigning a unique user name and strong password. Then you can safely disable the user "admin", to help prevent unauthorized access using this "well known" account. For related details, see "Configure any needed local users".

Continue to work in the station with admin write privileges on the new LDAP user service, in order to configure any needed local users, and configure the service for access by LDAP users on the network.

Configure any needed local users

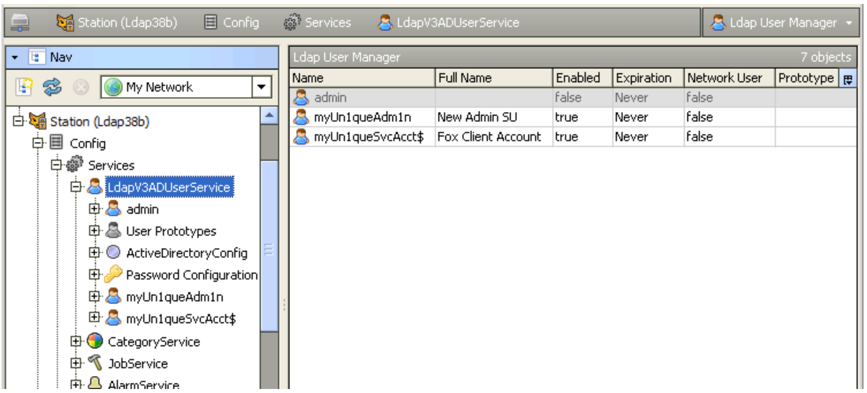
Typically, after you configure the LDAP user service and have it working, most user access to a station will be from LDAP users, that is, station login by users supplying their LDAP credentials. User components for these users will be dynamically created in the station—if they do not already exist.

First, however, you typically create a few *local* station users. These are in addition to the two "built-in" local users (admin and guest[1]) under any NiagaraAX user service. Station access by local users does not involve (nor is dependent upon) on LDAP server communications.

Typical use cases for additional local users are:

- To create a super user to *use instead* of the built-in (and well known) user "admin", such that you can set user "admin" to be *disabled*. Create this new super user *before disabling* the user admin.
- To create a special user to use as a "service account" for Fox station-to-station communications. Typically you assign admin write privileges to this user, and never use this for (person) login access to the station. Instead, you reference this user in *other (remote) stations*, when configuring the "Client Connection" properties under the **`NiagaraStation`** that represents this station.
  Note: *Although "in theory" an LDAP-sourced user could be used for a service account, we recommend creating and using a local user instead. This allows Niagara Network operations to continue even in the event of LDAP server issues.*

Figure 1-3 *Example local users (two) created in station with LdapV3ADUserService*



Name	Full Name	Enabled	Expiration	Network User	Prototype
admin		false	Never	false	
myUn1queAdm1n	New Admin SU	true	Never	false	
myUn1queSvcAcct$	Fox Client Account	true	Never	false	

Figure 1-3 above shows an example of two local users added to the LdapV3ADUserService:

[1] Starting in AX-3.8, the built-in local user "guest" is hidden after a station first starts. For security reasons, retaining this is the recommended configuration.

- `myUn1queAdm1n` — A user given "super user" privileges, allowing the user admin to be disabled.
- `myUn1queSvcAcct$` — A user with all admin-write privileges, used for station (Fox) client access from other stations.

To configure local users

Step 1 With the station opened in Workbench, double-click the user service for the **`Ldap User Manager`**.

Step 2 Click **`New`** to add one or more local users.

Step 3 In the **`New`** dialog for the user(s), configure properties as needed.

Step 4 **`Save`** the users and the station (right-click its **`Config`** node, and select **`Actions > Save`**).

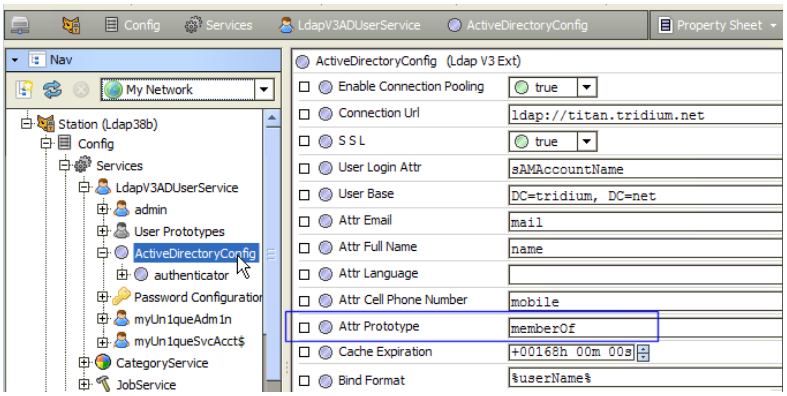
Notes on local users in an LDAP user service

- Typically after making a local user for "service account" station-to-station connections, you immediately update other stations in the Niagara Network such that they have the proper credentials in the "Client Connection" properties of the **`NiagaraStation`** (device) component that represents this station. For related details, see "About client connection properties" in the *Drivers Guide*.
- Properties accessible on the property sheet of the LDAP user service, for example to configure Password Strength and Lock Out-related settings, apply to all *local users only*, and not LDAP users. This also applies to "Password Configuration" user properties—that is, force reset at next login, expiring passwords, and password history mechanism. For LDAP users, such things are configured in the LDAP server/system, and not Niagara. For details on such local user functions, see the *User Guide* sections "Strong password notes", "Lockout notes", and "About password expiration and reset".
- "Network users" are not applicable—this function applies only if using the standard **`UserService`**, in which case UserPrototypes are used in a different manner than in an LDAP user service.

Configure User Prototypes

UserPrototypes under an LDAP user service are needed to map the appropriate station permissions and other items to different LDAP users that access the station. This mapping *begins* with a specific LDAP attribute applied to all LDAP users, as configured in the "Attr Prototype" property of the **`LdapConfig`** or **`ActiveDirectoryConfig`** child of the LDAP user service.

Figure 1-4 *Default attribute in Active Directory for prototype mapping (Attr Prototype) is "memberOf"*



ActiveDirectoryConfig (Ldap V3 Ext)

□ Enable Connection Pooling	true
□ Connection Url	ldap://titan.tridium.net
□ S S L	true
□ User Login Attr	sAMAccountName
□ User Base	DC=tridium, DC=net
□ Attr Email	mail
□ Attr Full Name	name
□ Attr Language	
□ Attr Cell Phone Number	mobile
□ Attr Prototype	memberOf
□ Cache Expiration	+00168h 00m 00s
□ Bind Format	%userName%

In the LdapConfig container for the LdapV3UserService or LdapUserService, this Attr Prototype property has no default value (is *blank*). But as shown in Figure 1-4, in the ActiveDirectoryConfig (of the LdapV3ADUserService or ActiveDirectoryService) the default value for this property is: `memberOf`

This property should *always* be set to specify a particular LDAP attribute, typically one used to define different groups or "types" of users in the LDAP system. Then, you create additional UserPrototype components under the service's UserPrototypes container, being careful to *name* them to match possible *values* for that attribute (as assigned to LDAP users). For example, you may create three UserPrototypes, named Manager, Operator, and Engineer (in addition to the existing Default Prototype).

In each of the UserPrototypes (including Default Prototype), you assign different, appropriate permissions and other Niagara-specific items, e.g. Nav File, Facets, Default Web Profile, and Mobile Web Profile.

When an LDAP user accesses the station, their LDAP-sourced user information such as name, email address, and language is used in the User component (automatically created for them). The remaining User properties are sourced from a UserPrototype that "name matches" the string *value* of the LDAP attribute specified in the "Attr Prototype". For example, an Active Directory user with a "memberOf" value of "Manager" would have the permissions, Nav file, etc. from the UserPrototype named *Manager*.

In the case where no "name matching" UserPrototype is found, the permissions, Nav file, etc. are sourced from the *Default Prototype*. Note if you leave the "Attr Prototype" property blank (default in LdapConfig), *all LDAP users* will use the Default Prototype property values—regardless if other UserPrototypes exist.

For related details, see "Mapping of permissions and other preferences to LDAP users" on page 2-5.

To add and configure UserPrototypes

With the station opened in Workbench:

Step 1 In the Nav tree, expand the user service to see **`User Prototypes`**; expand again for **`Default Prototype`**.

Step 2 Open the property sheet of the **`Default Prototype`** (double-click it).

Step 3 Set appropriate values for the following User properties:

- Permissions (for the Default Prototype, this typically is a very *minimal set* of permissions)
- Facets
- Nav File
- Default Web Profile
- Mobile Web Profile

Step 4 **`Save`** the property values in the User Prototype.

Step 5 In the Nav tree, right-click the **`Default Prototype`**, and select **`Duplicate`**.

Step 6 In the popup Name dialog, rename from "`defaultPrototype1`" to a possible attribute value name, for example, "Manager", "Engineer", and so on, and click **`OK`**.

Repeat this (duplication) until you have the required number of uniquely-named User Prototypes.

Step 7 Double-click each duplicated User Prototype, and in the property sheet of each one, adjust (as necessary) the same properties as you did for the Default Prototype (Permissions, Nav File, and so on), saving each when done. See Step 3 and Step 4.

For example, you may assign more Permissions and a different Nav file and Default Web Profile to a User Prototype named "Manager" than you would one named "Operator".

Step 8 When done, **`Save`** the station (right-click its **`Config`** node, and select **`Actions > Save`**).

Notes on User Prototypes in an LDAP user service

- Essentially, User Prototypes under an LDAP user service operate differently than they do under the standard (baja) **`UserService`**, where typically only the **`Default Prototype`** matters. However, just as with the standard UserService, the **`Default Prototype`** is always used as the "template" for all User property values when creating any *local user* in the station.
  Note "non-default User Prototypes" under the baja UserService are utilized in "network user" scenarios; however, network users are *not supported* if using an LDAP user service.
- If an LDAP user has *multiple* values for the "Attr Prototype" attribute, the top-to-bottom *ordering* of associated User Prototypes in the station can determine that User's property values (including permissions, among other things).
  For example, in an Active Directory scenario, if a user has "memberOf" values including "Engineer", "Operator" and "Manager", when that user accesses the station, if under the **`UserPrototypes`** container the User Prototype named "Operator" is ordered at the *top*, properties like Permissions, Nav File and so are source from *that* User Prototype (and not from any other User Prototype, even if the name matches another attribute value). For more details on this, see "Mapping of permissions and other preferences to LDAP users" on page 2-5.
- If after an LDAP user accesses the station and a User component is created for them (using whatever source User Prototype component property values), if you subsequently *change* a property in that User Prototype, this change should be dynamically updated in the corresponding User component. For example, if you increase permissions in that User Prototype, the permissions in any User that originally sourced from that User Prototype should also update to match.

Configure the LdapConfig or ActiveDirectoryConfig child

Any user service from the `ldap` module (and `ldap` palette) has a child container with properties you must configure for your specific LDAP (or Active Directory) server.

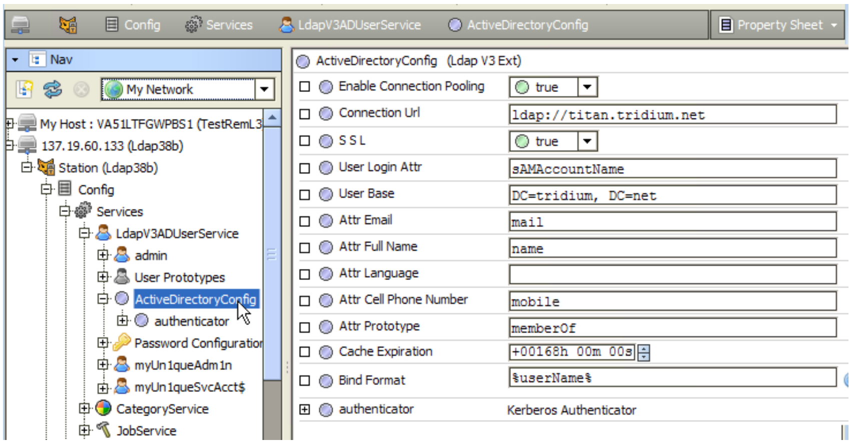
Depending on which user service you are configuring, this one of these components:

- **ActiveDirectoryConfig**
  If configuring the **`LdapV3ADUserService`** or **`ActiveDirectoryUserService`**
  see "Configuring the ActiveDirectoryConfig component" on page 1-8.
- **LdapConfig**
  If configuring the **`LdapV3UserService`** or **`LdapUserService`**
  see "Configuring the LdapConfig component" on page 1-9.

Configuring the ActiveDirectoryConfig component

With the station open in Workbench, expand the user service in the Nav tree and double-click on the **`ActiveDirectoryConfig`** node to access its property sheet.

Figure 1-5 *ActiveDirectoryConfig component in LdapV3ADUserService*



ActiveDirectoryConfig (Ldap V3 Ext)

□ Enable Connection Pooling	true
□ Connection Url	ldap://titan.tridium.net
□ S S L	true
□ User Login Attr	sAMAccountName
□ User Base	DC=tridium, DC=net
□ Attr Email	mail
□ Attr Full Name	name
□ Attr Language	
□ Attr Cell Phone Number	mobile
□ Attr Prototype	memberOf
□ Cache Expiration	+00168h 00m 00s
□ Bind Format	%userName%
⊞ authenticator	Kerberos Authenticator

Figure 1-5 above shows the ActiveDirectoryConfig container of the LdapV3ADUserService, with the default "authenticator" child (Kerberos Authenticator).

To configure the ActiveDirectoryConfig component

With the station opened in Workbench, in the **`ActiveDirectoryConfig`** component property sheet.

Step 1 As needed, change property values. Typical properties that may be changed from defaults are as follows:

- **Connection URL**
  URL of your LDAP (Active Directory) server, usually in the form: `ldap://your.domain.net`
  If the server uses a "non-standard" port, include in the URL, e.g. `ldap://your.domain.net:999` (note standard LDAP ports are 389, or else 636 if SSL).
  Note the scheme `ldaps://your.domain.net` is not supported in Connection URL.
- **SSL**
  Either 'false' (default) or 'true'. If 'true', the *station* uses SSL to communicate with the LDAP server.
  Note: *If true, be sure to enable SSL in the station's NiagaraNetwork's FoxService (for Workbench-to-station access) and also station's WebService (for browser-to-station access).*
- **User Login Attr**
  The specific attribute in the LDAP directory for the desired user login name.
  Note: *For Active Directory, the default* `sAMAccountName` *value is always used.*
- **User Base**
  Sub-tree of the LDAP server in which users who can access this station can be found. At the very least, it must contain the domain components of the server's domain, e.g. `DC=`*`domain`*`, DC=net`
- **Attr Email**
  The specific attribute in the LDAP directory to store user's email address, the value of which populates the Niagara user's Email property. The Active Directory default value is: `mail`
- **Attr Full Name**
  The specific attribute in the LDAP directory to store user's full name, the value of which populates the Niagara user's Full Name property. The Active Directory default value is: `name`
- **Attr Language**

The specific attribute in the LDAP directory to store user's language, the value of which populates the Niagara user's Language property. There is no default value (is blank).

- **Attr Cell Phone Number**
  The specific attribute in the LDAP directory to store user's cell phone number, the value of which populates the Niagara user's Cell Phone Number property. The default value is: `mobile`
- **Attr Prototype**
  The specific attribute in the LDAP directory to use for mapping a *User Prototype* (under the user service's **`UserPrototypes`** container) to users. The Active Directory default value is: `memberOf`
  This mechanism uses an "attribute value"-to-"component name" *matching* method of selection, where if no "name-matching" User Prototype is found, the frozen Default Prototype is used (when making the User component for the LDAP user, upon initial station login).
  For related details, see "Configure User Prototypes" on page 1-6.
- **Bind Format**
  (ActiveDirectoryConfig under LdapV3ADUserService only) If *not* using Kerberos, but instead the SimpleAuthenticator, it may be necessary to specify the *exact format* of the login name to send to the LDAP server. This can differ according to the LDAP server, and may be required more often when the "Authentication Choice" in the SimpleAuthenticator is DIGEST MD5. In some cases, just the user base and login name may be sufficient to find a user in the LDAP directory.
  For more information on this property, along with any other properties omitted above, see "ActiveDirectoryConfig (V3)" on page 2-12.

Step 2 **`Save`** the property values.

Step 3 In the LdapV3ADUserService, choose and configure a child authenticator. The default authenticator type is KerberosAuthenticator. Alternatively, you can use a SimpleAuthenticator. For details, see "Configure the LDAP authenticator (LdapV3 only)" on page 1-11.

Note: *In the (LDAPv2-compatible only) ActiveDirectoryService, a child authenticator component is not used. Instead, another property is used, found at the bottom of the ActiveDirectoryConfig property sheet:*

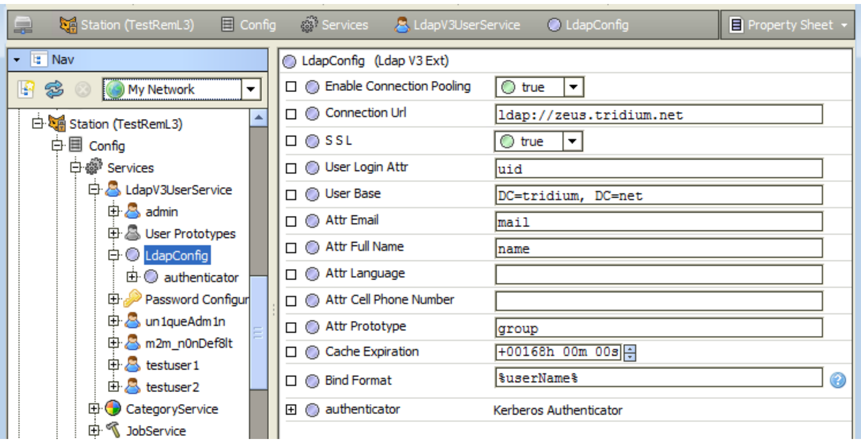
- **Domain**
  The value of this property is combined with the user's login name when authenticating against the server. For example, if the Domain property value is "`example.com`" and User Login Attr property value is "`sAMAccountName`", the ActiveDirectoryService would attempt to authenticate `janedoe` as `jandoe@example.com`.

For more details on the attribute ("Attr") properties, see "Attribute properties in LdapConfig" on page 2-4.

Configuring the LdapConfig component

With the station open in Workbench, expand the user service in the Nav tree and double-click on the **`LdapConfig`** node to access its property sheet.

Figure 1-6 LdapConfig component in LdapV3UserService



LdapConfig (Ldap V3 Ext)

□ Enable Connection Pooling	○ true ▾
□ Connection Url	ldap://zeus.tridium.net
□ S S L	○ true ▾
□ User Login Attr	uid
□ User Base	DC=tridium, DC=net
□ Attr Email	mail
□ Attr Full Name	name
□ Attr Language	
□ Attr Cell Phone Number	
□ Attr Prototype	group
□ Cache Expiration	+00168h 00m 00s
□ Bind Format	%userName%
⊞ authenticator	Kerberos Authenticator

Figure 1-6 above shows the LdapConfig container of the LdapV3UserService, with the default "authenticator" child (Kerberos Authenticator).

To configure the LdapConfig component

With the station opened in Workbench, in the **`LdapConfig`** component property sheet.

Step 1 As needed, change property values. Typical properties changed include the following:

- **Connection URL**
  URL of your LDAP (Active Directory) server, usually in the form: `ldap://your.domain.net`
  If the server uses a "non-standard" port, include in the URL, e.g. `ldap://your.domain.net:999` (note standard LDAP ports are 389, or else 636 if SSL).
  Note the scheme `ldaps://your.domain.net` is not supported in Connection URL.
- **SSL**
  Either 'false' (default) or 'true'. If 'true', the *station* uses SSL to communicate with the LDAP server.
  Note: *If true, be sure to enable SSL in the station's NiagaraNetwork's FoxService (for Workbench-to-station access) and also station's WebService (for browser-to-station access).*
- **User Login Attr**
  The specific attribute in the LDAP directory for the desired user login name.
  Note: *Different LDAP servers use a different attribute. For OpenLDAP, the attribute is:* `uid`
- **User Base**
  Sub-tree of the LDAP server in which users who can access this station can be found. At the very least, it must contain the domain components of the server's domain, e.g. `DC=`*`domain`*`, DC=`*`net`*
- **Attr Email**
  The specific attribute in the LDAP directory to store user's email address, the value of which populates the Niagara user's Email property. There is no default value (is blank).
- **Attr Full Name**
  The specific attribute in the LDAP directory to store user's full name, the value of which populates the Niagara user's Full Name property. There is no default value (is blank).
- **Attr Cell Phone Number**
  The specific attribute in the LDAP directory to store user's cell phone number, the value of which populates the Niagara user's Cell Phone Number property. There is no default value (is blank).
- **Attr Prototype**
  The specific attribute in the LDAP directory to use for mapping a *User Prototype* (under the user service's **`UserPrototypes`** container) to users. There is no default value (is blank).
  This mechanism uses an "attribute value"-to-"component name" *matching* method of selection, where if no "name-matching" User Prototype is found, the frozen Default Prototype is used (when making the User component for the LDAP user, upon initial station login).
  For related details, see "Configure User Prototypes" on page 1-6.
- **Bind Format**
  (LdapConfig under LdapV3UserService only) If *not* using Kerberos, but instead the SimpleAuthenticator, it may be necessary to specify the *exact format* of the login name to send to the LDAP server. This can differ according to the LDAP server, and may be required more often when the "Authentication Choice" in the SimpleAuthenticator is DIGEST MD5. In some cases, just the user base and login name may be sufficient to find a user in the LDAP directory.
  For more information on this property, along with any other properties omitted above, see "LdapConfig (V3)" on page 2-14.

Step 2 **`Save`** the property values.

Step 3 In the LdapV3UserService, choose and configure a child authenticator. The default authenticator type is KerberosAuthenticator. Alternatively, you can use the "SimpleAuthenticator". For more details, see "Configure the LDAP authenticator (LdapV3 only)" on page 1-11.

Note: *In the (LDAPv2-compatible only) LdapUserService, a child authenticator component is not used. Instead, there are two other properties are at the bottom of the LdapConfig property sheet:*

- **Connection User**
  The user name for the initial LDAP server connection. It may be required if users who will be logging in are in different sub-trees of the LDAP directory. If the LDAP server supports anonymous connections, leave this property empty (blank).
- **Connection Pwd**
  The password for the user specified in property Connection User.

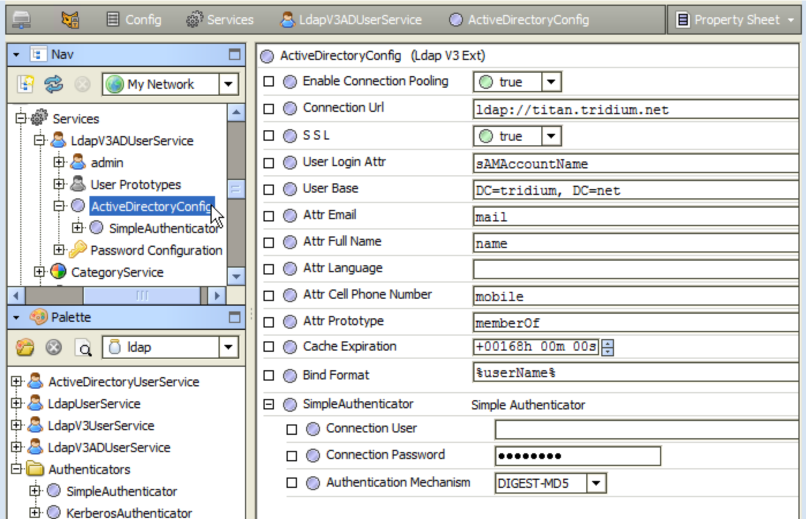
Configure the LDAP authenticator (LdapV3 only)

The newer LdapV3ADUserService and LdapV3UserService offer two main authentication choices:

- **SimpleAuthenticator**
  If using the LdapV3ADUserService or LdapV3UserService but *not using Kerberos* authentication, you can use the Simple Authenticator. See "Configuring the SimpleAuthenticator component".
  Note: *Use this if a "J9 Java VM" (JACE-2/4/5 series) station, as Kerberos is not supported.*
- **KerberosAuthenticator**
  This is the default authenticator for the **LdapV3ADUserService** or **LdapV3UserService**. See "Configuring the Kerberos Authenticator component" on page 1-12.

Configuring the SimpleAuthenticator component

Figure 1-7 *ActiveDirectoryConfig example using Simple Authenticator*



ActiveDirectoryConfig (Ldap V3 Ext)

Enable Connection Pooling	○ true
Connection Url	ldap://titan.tridium.net
SSL	○ true
User Login Attr	sAMAccountName
User Base	DC=tridium, DC=net
Attr Email	mail
Attr Full Name	name
Attr Language	
Attr Cell Phone Number	mobile
Attr Prototype	memberOf
Cache Expiration	+00168h 00m 00s
Bind Format	%userName%
SimpleAuthenticator	Simple Authenticator
Connection User	
Connection Password	••••••••
Authentication Mechanism	DIGEST-MD5

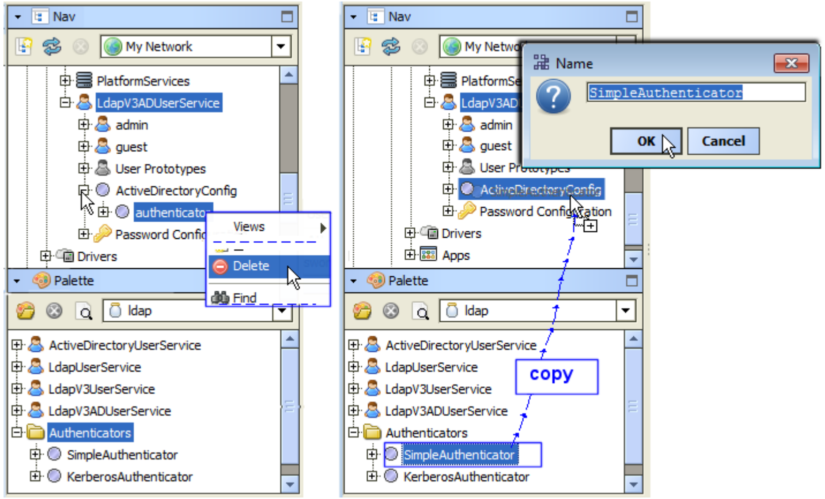
Figure 1-7 above shows an LdapV3ADUserService using the Simple Authenticator.

To configure the SimpleAuthenticator component

Step 1 With the station open in Workbench, expand the user service in the Nav tree and double-click on the **ActiveDirectoryConfig** node or **LdapConfig** node to access its property sheet.

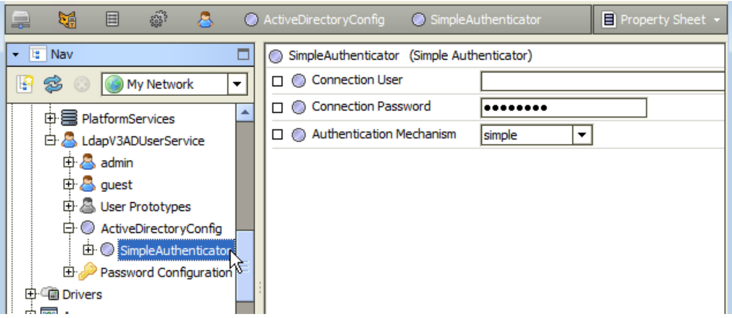
- If the authenticator is the "Kerberos Authenticator", *delete it* (right-click it and select **Delete**).
- If the authenticator is the "Simple Authenticator", go to Step 4.

Step 2 Open the `ldap` palette, and expand the **Authenticators** folder.



Step 3 In the Nav tree, copy (drag and drop) the **SimpleAuthenticator** from the palette onto the station's **ActiveDirectoryConfig** or **LdapConfig** component. Accept the default name in the popup **Name** dialog, or else enter another name, and click **OK**.

Step 4 Double-click the SimpleAuthenticator to configure its properties, as follows.



SimpleAuthenticator (Simple Authenticator)

□ Connection User	
□ Connection Password	••••••••
□ Authentication Mechanism	simple

- **Connection User**
  The user name for the initial connection. It may be required if the LDAP users logging in are in different sub-trees of the LDAP directory. If the LDAP server supports anonymous connections, this may not be required and can be left blank.
- **Connection Password**
  Password for the user specified as the "Connection User" (if applicable).
- **Authentication Mechanism**
  Specifies which SASL mechanism the station uses to authenticate to the server. Choices are:
  - simple — (default) Password is passed in clear text in authentication.
  - DIGEST-MD5 — Password is encrypted using Digest-MD5 mechanism.
  - CRAM-MD5 — Password is encrypted using CRAM-MD5 mechanism.
  - None — No authentication used.

  Note: *It is strongly recommended to use SSL when using simple authentication. Even though credentials in clear text will be used, at least it will be over an SSL connection.*

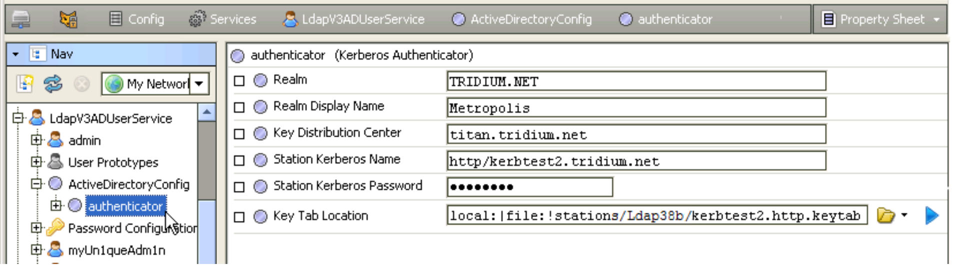
Configuring the Kerberos Authenticator component

Configuration requires prerequisite Kerberos information. See the "System requirements" subsection: "Kerberos prerequisites" on page 1-2.

To configure the KerberosAuthenticator component

Step 1 With the station open in Workbench, expand the user service in the Nav tree, including the **`ActiveDirectoryConfig`** node or **`LdapConfig`** node to see its child **`authenticator`**.

Step 2 Double-click the KerberosAuthenticator to configure its properties, as follows.



authenticator (Kerberos Authenticator)

□ Realm	TRIDIUM.NET
□ Realm Display Name	Metropolis
□ Key Distribution Center	titan.tridium.net
□ Station Kerberos Name	http/kerbtest2.tridium.net
□ Station Kerberos Password	••••••••
□ Key Tab Location	local:|file:!stations/Ldap38b/kerbtest2.http.keytab

- **Realm**
  The Kerberos realm on which the LDAP server resides, usually in all UPPERCASE letters, for example "`EXAMPLE.COM`". Typically, you get this information from the Kerberos administrator.
- **Realm Display Name**
  (usage optional) This field is blank by default. When accessing the station from a web browser, any text entered here replaces the "Realm" text in the lower "SSO area" of the station login dialog. For an example that matches the properties above, see Figure 2-7 on page 7.
- **Key Distribution Center**
  Name of the Kerberos Key Distribution Center, from which Kerberos users must contact to get tickets, for example "`kdc.example.com`". Again, you typically get this from the Kerberos administrator. Also see "Kerberos authentication notes" on page 1-13 about DNS considerations.
- **Station Kerberos Name**
  As part of securely delegating Kerberos tickets, the station must be a user in the Kerberos database, where this field represents the station in Kerberos. If logging in only via Workbench, this user can be any user or service in the Kerberos directory.

However, if logging in via a browser, the user must be a *service* in the form "`HTTP/serviceName.domain.com`", where "`serviceName.domain.com`" is how the station is to be accessed in the browser, (e.g. `http://jacekerb1.mydomain.com`).
The *service name* for station Kerberos name typically omits a bit of the normal http URL syntax, for example: `http/jacekerb1.mydomain.net` *instead* of `http://jacekerb1.mydomain.net`
You may need to ask the Kerberos administrator to create the service for you in the Kerberos database.

- **Station Kerberos Password**
  The password for the Kerberos station user specified by the "Station Kerberos Name". If using a keytab file, you can leave this blank (default).
- **KeyTab Location**
  Kerberos services typically do not use a password to authenticate. Instead, they use a file containing a key table (keytab file). If you want authentication from a web browser, you must specify an associated service in the "Station Kerberos Name" property, and reference a *keytab file* supplied by the Kerberos administrator.
  Copy that keytab file to a secure location on the NiagaraAX platform, somewhere under the *station's file space*. For example, copy it into the root of the station's file space. This in this KeyTab Location property field, use the **`File Ord Chooser`** to browse to it for selection. Again, if using a keytab, you can leave the "Station Kerberos Password" property value blank (default).

Kerberos authentication notes

- Kerberos is very particular about names. You must enter the station name in the "Station Kerberos Name" property *exactly* as it appears in the Kerberos database. Upper/lowercase can sometimes be an issue, so make sure you have an exact match.
- Kerberos uses "reverse DNS" to find the referenced Key Distribution Center. Therefore, it is essential to have a reverse DNS entry on *both the client and station*'s DNS servers.
  Otherwise, you will not be able to acquire Kerberos tickets, and you will not be able to log in. Contact the IT administrator to see if the appropriate entry exists on the server.
  Note: *As an alternative to having proper reverse DNS entry, you may also configure the* `hosts` *file on client PCs and station host(s) to map the IP address of the Key Distribution Center to its name.*

  For example, if the Key Distribution Center's name is `kdc.domain.net`*, and its IP address is* `123.156.78.90`*, add the following line to the hosts file:*

  ```
  123.156.78.90           kdc.domain.net
  ```

  On Windows PCs, the `hosts` *file is located at* `C:\Windows\System32\drivers\etc\hosts` *and on Linux hosts at* `/etc/hosts`*. On JACE platforms, use the platform TCP/IP Configuration view (or equivalent view on the station's TcpIpPlatformService) to access/edit the* `hosts` *file.*

  Finally, while modifying the hosts file is simple enough for a single station, and can be useful for testing your Kerberos setup, this approach can be tedious when dealing with multiple stations and multiple client machines. Setting up DNS servers with reverse DNS entries is the best option, if available.
- Login access to a Kerberos/LDAP station typically presents a different login dialog than when using the regular UserService. For operation details, see "Kerberos and NiagaraAX login operation" on page 2-5.

Additional Kerberos client-side setup

In addition to the NiagaraAX station (and station host) setup for Kerberos described in "Configuring the Kerberos Authenticator component" on page 1-12, there is "client-side" Kerberos setup required too. This additional setup applies to all computers that need to access (as a client) any station with a Kerberos authenticated LDAP user service, whether using either Workbench or web browser access.

Two different types of additional Kerberos-related client setup may be required:

- "Browser-independent setup" on page 1-14
- "Browser-specific setup" on page 1-15

Note: *In AX-3.8, apart from any Kerberos-related configuration, any browser client PC that needs to access a station to run "Web Workbench" (WbApplet), also requires "Unlimited Strength Policy Files" added to its Java installation. For details, see the section "Additional AX-3.8 client-side Java installation steps" in the latest* NiagaraAX 2013 Security Updates *document.*

Browser-independent setup

Verify/set up on each Workbench (or browser client) PC: a krb5.conf file and a Windows registry change.

krb5.conf

`krb5.conf` is a (text) configuration file for using Kerberos. It specifies items like the default realm to contact, or what flags should be set on tickets you acquire. In order to use Kerberos authentication with a NiagaraAX LDAP user service, you *must* be able to acquire *forwardable* Kerberos tickets.

To ensure this, you may need to edit your local `krb5.conf` file.

On a Windows host, you may find this file at:

`c:\winnt\krb5.ini` or else `c:\windows\krb5.ini` (note the different `.ini` extension)

Or on a Linux host, find this file at: `/etc/krb5.conf`

In any case, the “libdefaults” section in this file needs the following line:

```
    forwardable=true
```

If this file does not have this section, add the following lines at the top of the file:

```
    [libdefaults]
    forwardable=true
```

Save this file after making any change.

Note: *The following applies to `krb5.conf` file changes:*

- Some systems may require a more advanced `krb5.conf` file than the one discussed above. In that case, you typically have the Kerberos administrator set it up for you, if it does not already exist.
- If you do not have a `krb5.ini` or `krb5.conf` file, create it at one of the locations mentioned above. Alternatively, in any OS, you can create a `krb5.conf` file (note the file extension) in the directory:
  `<java_home>\lib\security` (Windows)
  `<java_home>/lib/security` (Linux and Solaris)
  This new file requires only the two lines previously shown.

Windows registry change

If using a Windows PC running Windows XP SP2 or higher, and would like to access your native Kerberos ticket, you need to set a registry key to allow Java access to it.

Note: *It is recommended to* backup *your Windows registry* before *making any changes. For more information, search on “Backing Up Your Registry” for instructions specific to your Windows version.*

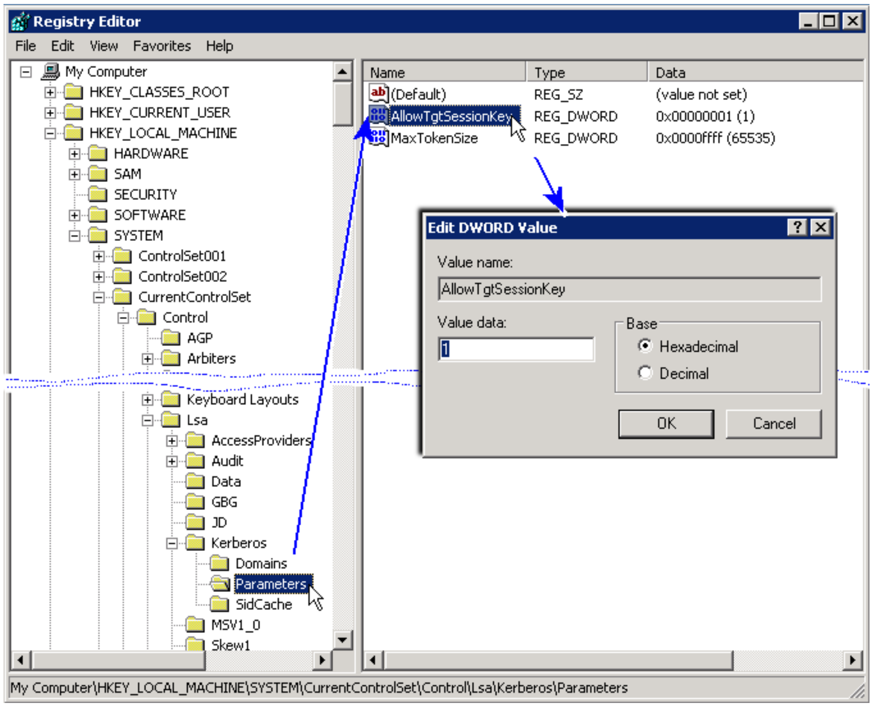
To set this registry key, start the registry editor (**`Start > Run...`**regedit) and add/edit the following key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters

    Value name: AllowTgtSessionKey
    Value type: REG_DWORD
    Value: 0x01
```

If using Windows XP, you should add/edit the key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos

    Value name: AllowTgtSessionKey
    Value type: REG_DWORD
    Value: 0x01
```

Figure 1-8 *AllowTgtSessionKey in Windows registry with required value*



Name	Type	Data
(Default)	REG_SZ	(value not set)
AllowTgtSessionKey	REG_DWORD	0x00000001 (1)
MaxTokenSize	REG_DWORD	0x0000ffff (65535)

Value name: AllowTgtSessionKey

Value data: 1

● Hexadecimal ○ Decimal

My Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters

Note: *If ever necessary, you can return to the default Windows security settings by changing the value of this registry key to 0.*

Browser-specific setup

In order to use Kerberos authentication in *browser access* of a station, a few things need to be configured on your local (browser) client PC (in *addition* to the "Browser-independent setup").

First, make sure you can connect to the station using a fully qualified domain name, rather than an IP address. When authenticating via Kerberos via a browser, the browser assumes that if you are on an address "`http://some.domain.com/somepage`", then you are trying to access the service "`HTTP/some.domain.com`" in the Kerberos database.

Since Kerberos processes names and not IPs, your IP must be mapped to the name of the service that is intended to use. If the correct entry is already in your DNS server, you do not need to do anything additional. If not, you can edit your client PC's `hosts` file to add an entry similar to:

"*`nnn.nnn.nnn.nnn   some.domain.com`*"

For example,

```
172.16.10.10   kerbtest2.tridium.net
```

Where the IP address above maps to the Kerberos service associated with kerbtest2 on tridium.net.

Note that hosts file method is acceptable for testing, but not so good once the site is live and many people need to access it.

Next, you'll need to set up your browsers to use Kerberos. As each browser is different, a separate procedure is given for the following popular browsers:

- Firefox
- Internet Explorer
- Chrome

Firefox

Using Firefox for browser access (as a Kerberos authenticated LDAP user) to a NiagaraAX station requires some client side setup. For prerequisites, see "Browser-independent setup" on page 1-14 and "Browser-specific setup" on page 1-15.

Configuring Firefox to use Kerberos for LDAP

To configure Firefox on a client LDAP host to use Kerberos:

Step 1 Open a Firefox window.

Step 2 Type “`about:config`” in the location bar and press Enter.

If a warning appears, continue on (“Promise to be careful”).

Step 3 In the Search box near the top of the page, type “`negotiate`”. This filters to six or seven attributes.

Step 4 Edit the following two entries:

- `network.negotiate-auth.delegation-uris`
- `network.negotiate-auth.trusted-uris`

Set each to include the uris of the station(s) you will be accessing via the browser, using a comma to separate if multiple stations.

For example, if accessing stations using the URLs `http://`*`host1.domain.com/somepage`*, and `http://`*`host2.domain.com/somepage`*, enter the value in these fields to:

`host1.domain.com,host2.domain.com`

Firefox should now be ready for Kerberos authentication. You should now be able to log in to stations without being prompted for a username and password.

Note: *At the time of this document, only Windows-based stations (Supervisor, AX SoftJACE, or JACE-NXT) or Linux-based stations support the “SSO login feature” from a browser. QNX-based JACE stations do not, and so require LDAP user login. For related details, see “Browser access via Kerberos” on page 2-7.*

Internet Explorer

Using Internet Explorer for browser access (as a Kerberos authenticated LDAP user) to NiagaraAX stations requires some client side setup. For prerequisites, see “Browser-independent setup” on page 1-14 and “Browser-specific setup” on page 1-15.

Configuring Internet Explorer to use Kerberos for LDAP

To configure Internet Explorer on a client LDAP host to use Kerberos, you must change security settings.

Step 1 Open an Internet Explorer window.

Step 2 On the menu bar, go to **`Tools > Internet Options`**

Step 3 In the **`Internet Options`** dialog, click the **`Security`** tab, and select the **`Local intranet`** zone.

Step 4 Click the **`Sites`** button, and in the popup click **`Advanced`**.

Another popup appears to “Add a website to this zone”.

Step 5 Type in the URL for a station and click **`Add`**.

For example, `http://`*`host1.domain.com`*

or, for SSL use (if HTTPS is enabled), `https://`*`host1.domain.com`*

If multiple stations, repeat each time by typing in the URL and clicking **`Add`**.

Step 6 When done adding stations, click **`Close`** then **`Ok`** to return to the **`Security`** tab.

Step 7 With the **`Local intranet`** zone selected, click the **`Custom level...`** button.

A popup **`Security Settings - Local intranet`** dialog appears.

Step 8 Scroll down to the “**`User Authentication`**” section (near the bottom), and select “Automatic logon only in Intranet zone” to use Kerberos authentication without a prompt. If you prefer to be prompted, select the option to “Prompt for user name and password”.

Step 9 Click **`OK`** twice to close the **`Internet Options`** dialog.

Internet Explorer should now be ready for Kerberos authentication. You should now be able to log in to stations without being prompted for a username and password.

Note: *At the time of this document, only Windows-based stations (Supervisor, AX SoftJACE, or JACE-NXT) or Linux-based stations support the “SSO login feature” from a browser. QNX-based JACE stations do not, and so require LDAP user login. For related details, see “Browser access via Kerberos” on page 2-7.*

Chrome

Using Google Chrome for browser access (as a Kerberos authenticated LDAP user) to NiagaraAX stations requires some client side setup. For prerequisites, see “Browser-independent setup” on page 1-14 and “Browser-specific setup” on page 1-15.

To configure Google Chrome on a client LDAP host to use Kerberos, you must change security settings to be like those on Internet Explorer, as well as specify additional startup arguments.

- Configuring Google Chrome security to use Kerberos for LDAP
  Note: *If you previously configured Internet Explorer to use Kerberos for LDAP access to NiagaraAX stations, this may already be done. However, the Chrome startup arguments still need configuration.*
- Configuring Google Chrome startup arguments

Configuring Google Chrome security to use Kerberos for LDAP

Step 1 Open a Google Chrome window.

Step 2 From the customize menu, select **`Settings`** (or type "`chrome:settings`" in the location bar and press Enter) for the "Chrome Settings" page. Click "**`Show advanced settings...`**" (near the bottom).

Step 3 Scroll down to the section **`Network`**, and click the **`Change proxy settings...`** button.

Step 4 In the **`Internet Options`** dialog, click the **`Security`** tab, and select the **`Local intranet`** zone.

Step 5 Click the **`Sites`** button, and in the popup click **`Advanced`**.

Another popup appears to "Add a website to this zone".

Step 6 Type in the URL for a station and click **`Add`**.

For example, `http://`*`host1.domain.com`*

If multiple stations, repeat each time by typing in the URL and clicking **`Add`**.

Step 7 When done adding stations, click **`Close`** then **`Ok`** to return to the **`Security`** tab.

Step 8 With the **`Local intranet`** zone selected, click the **`Custom level...`** button.

A popup **`Security Settings - Local intranet`** dialog appears.

Step 9 Scroll down to the "**`User Authentication`**" section (near the bottom), and select "Automatic logon only in Intranet zone" to use Kerberos authentication without a prompt. If you prefer to be prompted, select the option to "Prompt for user name and password".

Step 10 Click **`OK`** twice to close the **`Internet Options`** dialog.

Step 11 Close all Chrome windows. See Configuring Google Chrome startup arguments.

Configuring Google Chrome startup arguments

When you start Chrome (`chrome.exe`), you need the following two arguments appended:

```
--auth-negotiate-delegate-whitelist="host1.domain.com" " --auth-server-
whitelist="host1.domain.com"
```

Where the URL for the station(s) is in quotation marks as shown above. Note if multiple stations, use a comma to separate each one, as shown below.

```
--auth-negotiate-delegate-whitelist="host1.domain.com","host2.domain.com" " --
auth-server-whitelist="host1.domain.com","host2.domain.com"
```

If starting Chrome from a command line, append the arguments above to the end of the command.

If starting Chrome from a shortcut, do the following:

Step 1 Right-click the shortcut used to start Chrome and select "**`Properties`**".

Step 2 From the "**`Shortcut`**" tab, click in the "**`Target`**" field, and go to the end (click **`End`**).

Step 3 Append the arguments as shown above to the command (after any quotation marks that may already be there).

Step 4 Click **`OK`** to save the shortcut.

Google Chrome should now be ready for Kerberos authentication. You should now be able to log in to stations without being prompted for a username and password.

Note: *At the time of this document, only Windows-based stations (Supervisor, AX SoftJACE, or JACE-NXT) or Linux-based stations support the "SSO login feature" from a browser. QNX-based JACE stations do not, and so require LDAP user login. For related details, see "Browser access via Kerberos" on page 2-7.*

CHAPTER 2

NiagaraAX LDAP Concepts and References

The NiagaraAX `ldap` module contains different user service components, from which you can select to *replace* a station's standard UserService. Concepts about using these LDAP-sourced user services are in following main sections, along with reference details on related components and views:

LDAP server types

An LDAP (Lightweight Directory Access Protocol) server typically provides an IP network-accessible, hierarchical, distributed database, where an organization can store information about authorized users and their privileges. These distributed "directory services" can be used by many other hosts on the network, yet managed using a single point of administration.

Regarding integration with NiagaraAX, there are two basic types of LDAP server implementations:

- Active Directory
- LDAP

Active Directory

Active Directory is the Microsoft-supplied directory service used on many Windows domain networks, and is included in most Windows Server operating systems. AD (Active Directory) utilizes protocols LDAP (LDAPv2 or LDAPv3) and often Kerberos for authentication, and supports an LDAP interface. With Windows AD, users can access resources anywhere on the network with single logon.

The Windows AD structure uses a tree-type hierarchy of objects.

Because of the popularity of Windows domain networks, AD installations may be the most widely implemented LDAP systems. When integrating a Windows AD system with NiagaraAX stations, you replace the standard UserService in each station with one of the following user services:

- **ActiveDirectoryUserService**
  Only for LDAPv2-based ADs, and *without* availability of Kerberos authentication. Essentially this is the same AD-specific LDAP user service provided in the NiagaraAX `ldap` module since AX-3.1.
- **LdapV3ADUserService**
  For any LDAPv3-based AD, with availability of Kerberos authentication. Starting in AX-3.8, the `ldap` module provides this for all NiagaraAX platforms. The host platform must be licensed with the feature `"ldapv3"`. If Kerberos authentication is used, the `"ldapv3"` feature requires the attribute `kerberos="true"`.

LDAP

Apart from Window's AD (Active Directory), a number of other LDAP server implementations are in usage, including "open source" solutions such as Apache Directory Server and OpenLDAP. As with Windows AD, LDAPv2 or LDAPv3 may be used, and Kerberos authentication may be available. Such LDAP systems also use a tree-type directory structure of entries, each with sets of attributes.

When integrating any of these LDAP systems with NiagaraAX stations, you replace each station's standard UserService with one of the following user services:

- **LdapUserService**
  Only for an LDAPv2-based system, and *without* availability of Kerberos authentication. Essentially this is the same LDAP user service provided in the NiagaraAX `ldap` module since AX-3.1.
- **LdapV3UserService**
  For an LDAPv3-based system, with the availability of Kerberos authentication. Starting in AX-3.8, the `ldap` module provides this for all NiagaraAX platforms. The host platform must be licensed with the feature `"ldapv3"`. If Kerberos authentication is used, the `"ldapv3"` feature requires the attribute `kerberos="true"`.

LDAP and NiagaraAX operation

The following topics explain how LDAP and NiagaraAX operate together.

- NiagaraAX and LDAP interaction and benefits
- Local users versus LDAP users
- Attribute properties in LdapConfig
- Mapping of permissions and other preferences to LDAP users

NiagaraAX and LDAP interaction and benefits

Corporate or campus installations that already use Windows Active Directory or other LDAP-based "directory services" to manage user access across distributed networked resources can *benefit* from configuring NiagaraAX stations to use an LDAP user service. Benefits include:

- Automatic creation of station user accounts upon LDAP user login, with pre-determined permissions (set by UserPrototypes, as configured under the station's LDAP user service). LDAP-sourced users automatically reflect existing User property data like email address, full name, and language.

- Starting in AX-3.8, the ability to use Kerberos authentication for access of NiagaraAX stations by LDAP users, available for LDAPv3-based systems (Active Directory or other LDAP systems). Kerberos offers a high-level of security, albeit with some required client setup of hosts and browsers. Station login for Kerberos authenticated systems (either from Workbench or a client browser) offer choices for LDAP users to "log in as current user"—without any need to enter credentials, or alternatively to log in as a different user, providing credentials. This simplifies access for most users.
  Note: *Kerberos is not supported on any "J9 Java VM" (JACE-2/4/5 series) platform.*
- Bypass of any necessary NiagaraAX "network user" configuration, which is incompatible for any station configured with an LDAP user service. Thus, most central management of station users remains coordinated by of the installation's existing LDAP (AD) server.

Note: *There is no "hybrid" support for both LDAP users and the "network users" feature of a NiagaraNetwork. Therefore, all stations (both Supervisors and JACEs) on an LDAP-served installation require the "standard" UserService replaced by an LDAP user service.*

Local users versus LDAP users

A station with an LDAP user service (from the `ldap` module) can still have "local" NiagaraAX users. In fact, the standard two "frozen" user accounts (admin "super user" and guest user) are present in any of these user services, just as in the standard `baja` UserService component. More local users, if needed, can be added in the standard way—in the **`Ldap User Manager`** view, click the **`New`** button, and so on.

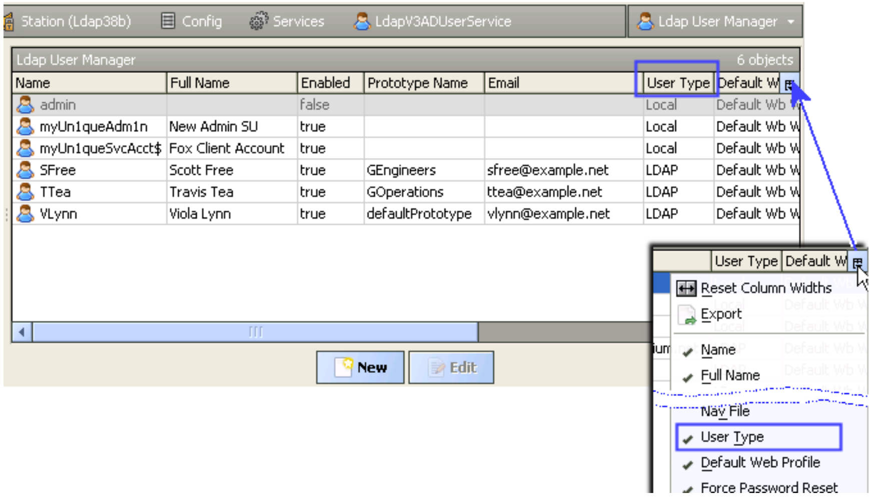
Typically, local station users are rarely used, except for the possible initial "super user" level configuration of the station. However, often an *additional* local user is added, to use exclusively as a NiagaraNetwork "service user". This service user would be referenced in other (remote) NiagaraAX stations, in the "Client-Connection" container of the NiagaraStation device that represent this station.

Although (in theory) an LDAP-sourced "service user" could be used, that is *not recommended*. Instead, create and use a *local user* as a service user. This makes the initial configuration of a NiagaraNetwork more straightforward. It also provides immunity from station-to-station communication issues that might arise, say from LDAP password expiration rules, or in the unlikely event of LDAP server problems.

Ldap User Manager view difference

The default view of any NiagaraAX LDAP user service is the **`Ldap User Manager`**.

Figure 2-1 Ldap User Manager view is default view on any LDAP user service



Name	Full Name	Enabled	Prototype Name	Email	User Type	Default W
admin		false			Local	Default Wb W
myUn1queAdm1n	New Admin SU	true			Local	Default Wb W
myUn1queSvcAcct$	Fox Client Account	true			Local	Default Wb W
SFree	Scott Free	true	GEngineers	sfree@example.net	LDAP	Default Wb W
TTea	Travis Tea	true	GOperations	ttea@example.net	LDAP	Default Wb W
VLynn	Viola Lynn	true	defaultPrototype	vlynn@example.net	LDAP	Default Wb W

This view is identical to the default **`User Manager`** view for the standard (baja) **`UserService`**, with only one addition: a "User Type" column (by default, selected to display). As shown in Figure 2-1, the value for each user is either "Local" or "LDAP".

Apart from this, the **`Ldap User Manager`** functions exactly the same as the standard (baja) **`User Manager`** view for the management of all local station users. This includes "Password Configuration" properties for local station users. For related details, see "UserService" in the NiagaraAX *User Guide.*

Note: *One exception to this is the "network user" function, which is not applicable to any users under any LDAP user service—either local station users or LDAP users.*

Attribute properties in LdapConfig

Several key configuration settings of any LDAP user service are “attribute” properties. Find them in the either the **`LdapConfig`** container or **`ActiveDirectoryConfig`** container (depending on type of user service). These properties correspond to *names* of specific *attributes* in the target *LDAP directory*.

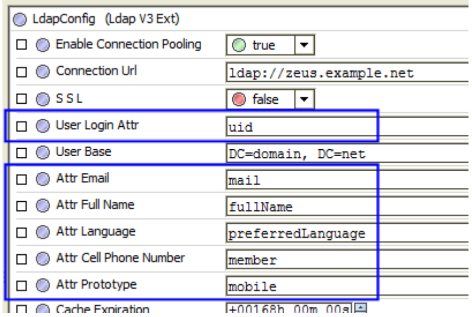
Each entry in the LDAP directory (say for a person or user) has multiple attributes which may or may not be assigned values. For example, a company LDAP directory might have entries for all employees, each with various attributes like “Full name”, “Phone number”, “Title”, “Address”, and so on.

A sample LDAP user entry may include attribute names and values as shown below.

```
User: jdoe
uid: jdoe
fullName: John Doe
title: Software Engineer
employeeNumber: E666
mobile: 555-555-0103
mail: jdoe@example.net
preferredLanguage: en
member: Engineering
```

In the station’s LdapConfig or ActiveDirectoryConfig attribute (`Attr`) properties, values should correspond to the names of the attributes in the LDAP directory. The value of the LDAP attribute is then pulled from the LDAP directory to fill out information about the user.

Figure 2-2 *Attr (attribute) properties in LdapConfig*



LdapConfig (Ldap V3 Ext)

□ Enable Connection Pooling: true
□ Connection Url: `ldap://zeus.example.net`
□ SSL: false
□ User Login Attr: `uid`
□ User Base: `DC=domain, DC=net`
□ Attr Email: `mail`
□ Attr Full Name: `fullName`
□ Attr Language: `preferredLanguage`
□ Attr Cell Phone Number: `member`
□ Attr Prototype: `mobile`
□ Cache Expiration: `+00168h 00m 00s`

For example, for the Niagara station user “jdoe” in the LDAP sample entry above to have a “Full Name” property value, you enter “`displayName`” in the “Attr Full Name” field. The attribute properties are:

- **User Login Attr**
  The specific attribute in the LDAP directory for the desired user login name.
  Note: *Different LDAP servers use different attributes. With OpenLDAP, the attribute is:* `uid` *while in Active Directory it is* `sAMAccountName`*.*
- **Attr Email**
  The specific attribute in the LDAP directory to store user’s email address, the value of which populates the Niagara user’s Email property.
- **Attr Full Name**
  The specific attribute in the LDAP directory to store user’s full name, the value of which populates the Niagara user’s Full Name property.
- **Attr Language**
  The specific attribute in the LDAP directory to store the user’s language, typically an ISO 639 two-letter language code, the value of which populates the Niagara user’s Language property.
- **Attr Cell Phone Number**
  The specific attribute in the LDAP directory to store user’s cell phone number, the value of which populates the Niagara user’s Cell Phone Number property.
- **Attr Prototype**
  The specific attribute in the LDAP directory to use for mapping a *User Prototype* (under the user service’s **`UserPrototypes`** container) to users.
  This mechanism uses an “attribute value”-to-“component name” *matching* method of selection, where if no “name-matching” User Prototype is found, the frozen DefaultPrototype is used (when making the User component for the LDAP user, upon initial station login).
  For related details, see “Configure User Prototypes” on page 1-6.

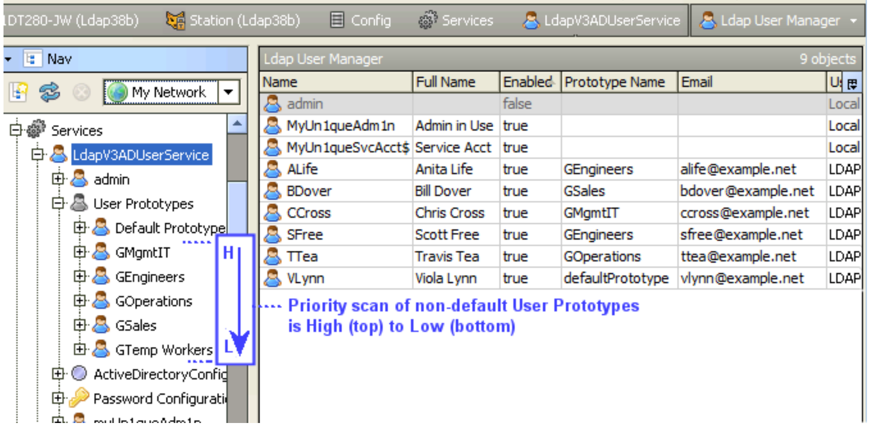
Mapping of permissions and other preferences to LDAP users

The first time an LDAP user accesses (logs into) a station, a User component is created in the station, named as that user's "user login name" on the LDAP server. Other user properties "Full Name", "Email", and "Language" are also sourced directly from the LDAP server.

Other properties of that User are supplied by a local "user prototype" in the station—that is, a User under the user service's **`UserPrototypes`** container. Permissions are the most critical of these items, which also include properties Nav File, Language, Facets, Default Web Profile, and Mobile Web Profile. For related details, see "Configure User Prototypes" on page 1-6.

Selection of *which* user prototype "to use as source" employs a "top-to-bottom" priority scan of all available Users under UserPrototypes. See Figure 2-3 below.

Figure 2-3 *Order of User Prototypes can determine which User Prototype is selected for LDAP user*



Ldap User Manager — 9 objects

Name	Full Name	Enabled	Prototype Name	Email	U
admin		false			Local
MyUn1queAdm1n	Admin in Use	true			Local
MyUn1queSvcAcct$	Service Acct	true			Local
ALife	Anita Life	true	GEngineers	alife@example.net	LDAP
BDover	Bill Dover	true	GSales	bdover@example.net	LDAP
CCross	Chris Cross	true	GMgmtIT	ccross@example.net	LDAP
SFree	Scott Free	true	GEngineers	sfree@example.net	LDAP
TTea	Travis Tea	true	GOperations	ttea@example.net	LDAP
VLynn	Viola Lynn	true	defaultPrototype	vlynn@example.net	LDAP

Priority scan of non-default User Prototypes is High (top) to Low (bottom)

Mapping (selection) is to the first User Prototype with a *name* that *matches* a *value* returned from the LDAP server for a specifc LDAP attribute (as specified in the "Attr Prototype" property, as in the user service's "LdapConfig" or "ActiveDirectoryConfig" container). For Active Directory, this is the "`memberOf`" attribute.

Note an LDAP user may be a member of *multiple* groups. In the example above, user CCross (Chris Cross) is a member of AD groups GEngineers, GOperations, and GMgmtIT. Because the User Prototype GMgmtIT is ordered at the top, when his User component is created in the station it uses *that* User Prototype for permissions, Nav file, and so on.

If no matching-named user prototype is found, the frozen "Default Prototype" User is the source for permissions and other User properties locally sourced in the station. In the example above, this is the case for user VLynn (Viola Lynn).

Kerberos and NiagaraAX login operation

Kerberos is an available authentication mechanism for LDAP starting in AX-3.8, if using either the **`LdapV3ADUserService`** or **`LdapV3UserService`** on any host running the "Hotspot" Java VM (recent JACE platforms and all Windows-based hosts). Configuration involves both NiagaraAX host and station configuration, as well as configuration of client hosts (and browsers) that will access the system. For related details, see the following sections:

- "Kerberos prerequisites" on page 1-2
- "Configuring the Kerberos Authenticator component" on page 1-12
- "Additional Kerberos client-side setup" on page 1-13

Once stations and client hosts are configured, login access behavior to a station varies as follows:

- "Workbench access via Kerberos" on page 2-5
- "Browser access via Kerberos" on page 2-7

Workbench access via Kerberos

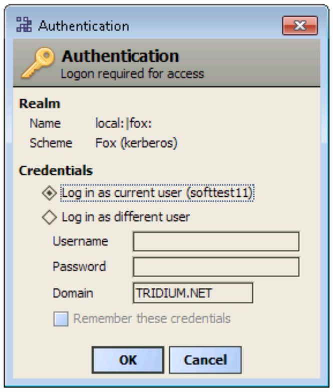
When using Workbench to access a station using LDAP/Kerberos, you see one of two login dialogs.

- Workbench login with found Kerberos credentials
- Workbench login without Kerberos credentials

Workbench login with found Kerberos credentials

If Workbench was able to aquire your native Kerberos credentials (see "Additional Kerberos client-side setup" on page 1-13), you see a dialog similar to Figure 2-4.

Figure 2-4 *Workbench login dialog if Kerberos credentials found*



Authentication

Authentication
Logon required for access

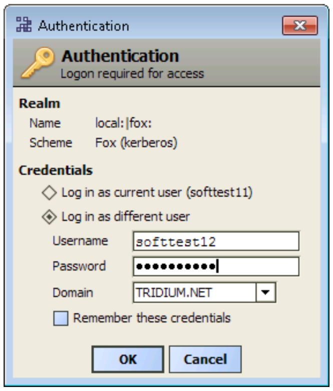
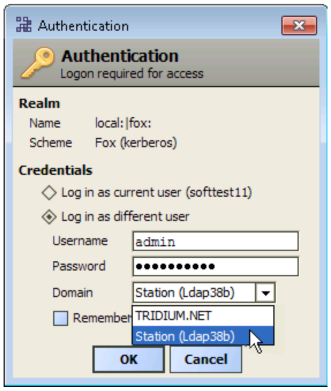
Realm
Name local:|fox:
Scheme Fox (kerberos)

Credentials
◆ Log in as current user (softtest11)
◇ Log in as different user
Username
Password
Domain TRIDIUM.NET
□ Remember these credentials

Here you can simply click **OK** to login as the current user.

Note: *If you want to log in as the current user, your client host must be part of the same realm as the station.*

Or, to login either as a different LDAP user or a local (station) user, select "Log in as different user", as shown below.

Log in as different LDAP user	Log in as local station user
 Authentication **Authentication** Logon required for access **Realm** Name local:|fox: Scheme Fox (kerberos) **Credentials** ◇ Log in as current user (softtest11) ◆ Log in as different user Username softtest12 Password •••••••••• Domain TRIDIUM.NET □ Remember these credentials OK Cancel	 Authentication **Authentication** Logon required for access **Realm** Name local:|fox: Scheme Fox (kerberos) **Credentials** ◇ Log in as current user (softtest11) ◆ Log in as different user Username admin Password •••••••••• Domain Station (Ldap38b) □ Remember TRIDIUM.NET Station (Ldap38b) OK Cancel

To log in as a different user:

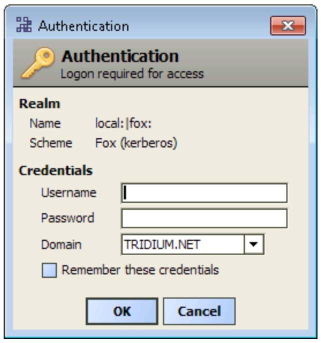
- If you want to log in as a *different LDAP user*, leave Domain at the realm, as above *left*.
- To log in as a local (station) user, change the Domain to "Station (*stationName*)", as above *right*.

Enter the appropriate username and password.

Workbench login without Kerberos credentials

If Workbench was unable to acquire your native Kerberos credentials, you see a simplified login dialog, similar to Figure 2-5.

Figure 2-5 *Workbench login dialog without found Kerberos credentials*



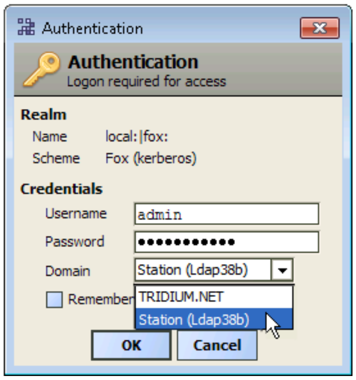
Realm
Name local:|fox:
Scheme Fox (kerberos)

Credentials
Username
Password
Domain TRIDIUM.NET
□ Remember these credentials

This dialog does not provide an option to log in as the current user. To log in, you must supply credentials.

- For *LDAP user* log in using Kerberos, leave Domain at the realm, and enter your credentials (LDAP user name and password).
- To log in as a local (station) user, change Domain to "Station (*stationName*)", as shown below in Figure 2-6, and enter your credentials.

Figure 2-6 *Workbench login as local station user when Kerberos credentials not found*



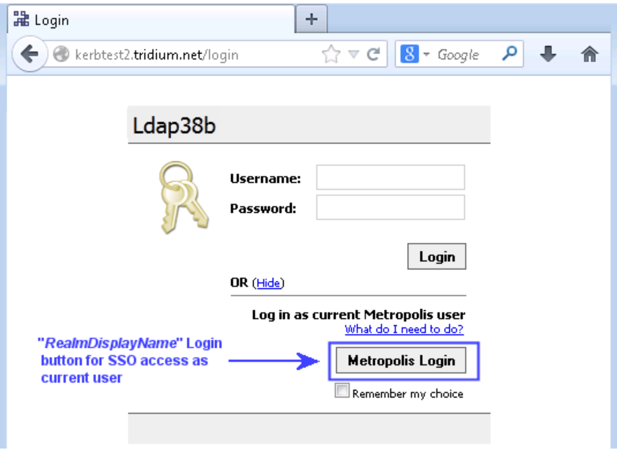
Realm
Name local:|fox:
Scheme Fox (kerberos)

Credentials
Username admin
Password ●●●●●●●●●●●
Domain Station (Ldap38b)
□ Remember

Browser access via Kerberos

If using a browser to access a station using LDAP/Kerberos, by default you may see a login dialog similar to Figure 2-7 below.

Figure 2-7 *Example default browser login to station with LDAP/Kerberos*



"*RealmDisplayName*" Login button for SSO access as current user

Often[1], you simply click this "***`RealmDisplayName Login`***" button to log in as the current user, without having to enter any credentials. Alternatively, for single sign-on (SSO) access you may go to the `/login-kerb` page (instead of the default `/login` page), whereby you are directly logged into the station (no button press required).

Note: *If you can successfully log in using single sign-on, and want to bypass the login dialog shown above in the future (logging directly into the station), check the "***`Remember my choice`***" check box at the bottom. This is effectively the same as going to the station's* `/login-kerb` *page.*

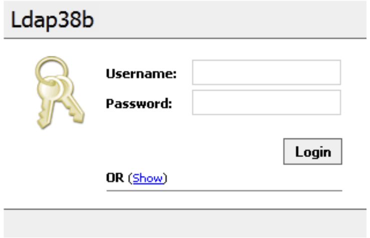
To subsequently remove the bypass *of the login dialog (from either technique above), clear your browser's cookies. Then, upon the next access from the browser, the station login dialog returns (Figure 2-7)*

Note in order to log in using your current LDAP credentials, the station must reside on the same realm as you. For example, if you are logged into the FACTORY realm, you will not be able to use your credentials to access a station set up for the EXAMPLE realm.

If the station is not set up to use the same realm as your currently-logged-in user, you can enter your Kerberos/LDAP credentials directly into the credentials fields and click the "**`Login`**" button. These are sent over to the station in plain text, where the station then takes care of the Kerberos authentication. In this case it is *strongly recommended* that you use SSL, so as to protect your password. See "Credentials access using browser" on page 2-9 for a related caution.

If not using the SSO login access feature, you can click the "`OR (`Hide`)`" link in the login dialog. In this case, the browser login dialog collapses to a smaller size (Figure 2-8).

Figure 2-8 Browser access login to station with LDAP/Kerberos, set to hide SSO login



Click the "`OR (`Show`)`" link to toggle the login dialog back full size, as shown in Figure 2-7 on page 2-7. The last used (`Show`, `Hide`) login dialog setting should remain cached in your browser.

See the following topics for more details on browser access to an LDAP/Kerberos station:

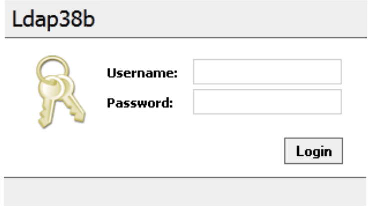
- Kerberos usage notes for JACE stations
- Credentials access using browser
- About Realm Display Name
- About the single sign-on access help link

Kerberos usage notes for JACE stations

Kerberos is supported on all "Hotspot JACEs", such as newer QNX-based JACE controllers (JACE-3/6/7 series). However, all QNX-based JACE controllers currently use Java5, which does not support Kerberos tokens sent by browsers for the SSO login feature.

Therefore, to access a JACE station using Kerberos via a browser, you must enter your LDAP credentials in the Username and Password fields, and use the regular "**`Login`**" button. The station login dialog from a browser appears similar to Figure 2-9.

Figure 2-9 Browser access login to JACE station with LDAP/Kerberos (normal login dialog)



As shown in Figure 2-9, the SSO login features (toggled with `Show` or `Hide`) do not appear in the login dialog to a QNX-based JACE station.

See "Credentials access using browser" for further details and a caution.

[1] If a QNX-based JACE station, the SSO features are not supported—instead you must enter credentials and use the normal Login button. See "Kerberos usage notes for JACE stations" on page 2-8 for background information, and also "Credentials access using browser" on page 2-9.

Credentials access using browser

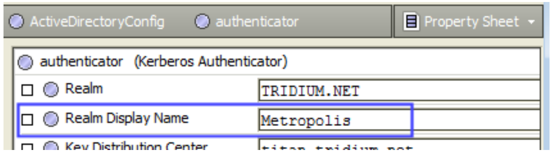
To log into the station as an LDAP user when SSO access is not supported, or to login as a different LDAP user (or a local station user, e.g. "admin"), enter the appropriate username and password in the credentials fields and click the upper "**`Login`**" button.

Caution *Anytime you chose to do this (Login entering credentials and the* **`Login`** *button), we strongly recommend that you use SSL, so as to protect your password.*

About Realm Display Name

Text for "*RealmDisplayName*" is set by a property value in the station's Kerberos authenticator, as shown in the Figure 2-10 example below. By default, this property is blank, and usage is optional.

Figure 2-10 *Realm Display Name value affects browser access login dialog*



authenticator (Kerberos Authenticator)

□ Realm TRIDIUM.NET

□ Realm Display Name Metropolis

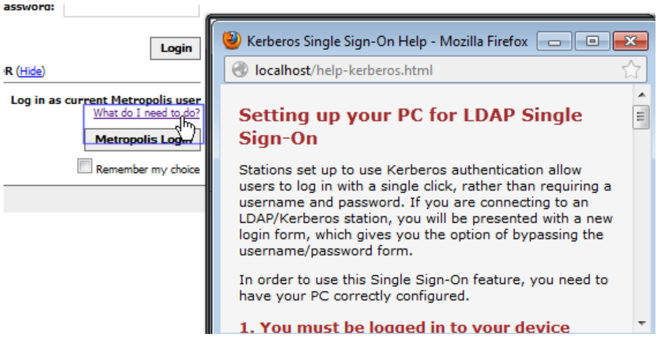
□ Key Distribution Center titan.tridium.net

- If used, this text appears in the full browser login dialog, as shown in Figure 2-7 on page 2-7, used for SSO access (no credentials required).
- If this property is left blank, the "***`RealmDisplayName`*** **`Login`**" button in that dialog shows "***`Realm`*** **`Login`**", e.g. "**`TRIDIUM.NET Login`**". The "Realm" text is also used above the button, for example "**`Log in as current TRIDIUM.NET user`**".

About the single sign-on access help link

The default browser login dialog includes a "`What do I need to do>`" link above the "***`RealmDisplayName`*** **`Login`**" button. This link produces a popup window with Kerberos-specific setup information summarized for the client PC and its different browsers, as shown in Figure 2-11.

Figure 2-11 *Kerberos single sign-on browser help link*


assword:
Login
R (Hide)
Log in as current Metropolis user
What do I need to do?
Metropolis Login
□ Remember my choice
Kerberos Single Sign-On Help - Mozilla Firefox
localhost/help-kerberos.html

Setting up your PC for LDAP Single Sign-On

Stations set up to use Kerberos authentication allow users to log in with a single click, rather than requiring a username and password. If you are connecting to an LDAP/Kerberos station, you will be presented with a new login form, which gives you the option of bypassing the username/password form.

In order to use this Single Sign-On feature, you need to have your PC correctly configured.

1. You must be logged in to your device

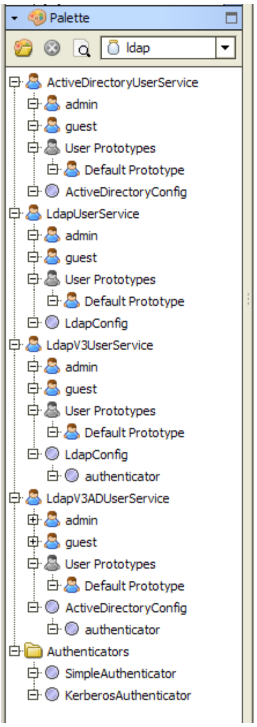
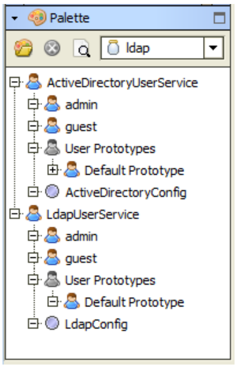
This information may be useful as a "first tier" check for an LDAP user to follow if the single sign-on (SSO) feature is not working. This same information is in this document's "LDAP / Active Directory Quick Start" section, mostly in the subsection "Browser-specific setup" on page 1-15.

Note: *Remember, the SSO feature from a browser does not work if accessing a station on a QNX-based JACE, only a station running on a Windows-based host. See "Kerberos usage notes for JACE stations" on page 2-8.*

ldap module palette

The `ldap` module's palette in AX-3.8 includes four different user services and two "authenticators", as shown in Figure 2-12 with all items expanded. This differs from the palette in AX-3.7 and earlier releases.

Figure 2-12 *Palettes for ldap module in AX-3.8 vs. AX-3.7*

AX-3.8 ldap palette	AX-3.7 ldap palette
 Palette ldap ActiveDirectoryUserService admin guest User Prototypes Default Prototype ActiveDirectoryConfig LdapUserService admin guest User Prototypes Default Prototype LdapConfig LdapV3UserService admin guest User Prototypes Default Prototype LdapConfig authenticator LdapV3ADUserService admin guest User Prototypes Default Prototype ActiveDirectoryConfig authenticator Authenticators SimpleAuthenticator KerberosAuthenticator	 Palette ldap ActiveDirectoryUserService admin guest User Prototypes Default Prototype ActiveDirectoryConfig LdapUserService admin guest User Prototypes Default Prototype LdapConfig

The top two user services (ActiveDirectoryUserService and LdapUserService) are the same components provided in AX-3.7 and earlier releases, and support LDAPv2 only. In AX-3.8, the *lower two* user services (LdapV3UserService and LdapV3ADUserService) and authenticators are *new*, and apply to systems using LDAPv3—either with or without Kerberos authentication.

Both the LdapV3UserService and LdapV3ADUserService are also *backwards-compatible* with an LdapV2 system. Each of these components is briefly described as follows:

Note: *Any of these user services is intended to replace a station's standard UserService component. Note either LdapV3 user service is a licensed feature, and Kerberos authentication (if used) is also a licensed feature.*

- **ActiveDirectoryUserService**
  (and children) The NiagaraAX station user service to support a Windows Active Directory (AD) service that uses LDAP version 2 (LDAPv2). The child ActiveDirectoryConfig component holds all properties needed to configure connection to the AD domain controller. For reference details, see "ActiveDirectoryUserService" on page 2-16.
- **LdapUserService**
  (and children) The NiagaraAX station user service to support an LDAP version 2-based LDAP server. The child LdapConfig component holds all properties needed to configure connection and authentication to the LDAP server. For reference details, see "LdapUserService" on page 2-17.
- **LdapV3UserService**
  (and children) The NiagaraAX station user service to support an LDAP version 3-based LDAP server (as well as an LDAP version 2-based server). The child LdapConfig component holds all properties needed to configure connection to the LDAP server, including an "authenticator" container for properties used to authenticate to it.
  By default, Kerberos authentication is used, applicable to LDAPv3 systems configured to use Kerberos authentication. If needed, after copying this user service in the station you can *replace* the default authenticator with the "SimpleAuthenticator" copied from the `ldap` palette. For reference details, see "LdapV3UserService" on page 2-13.

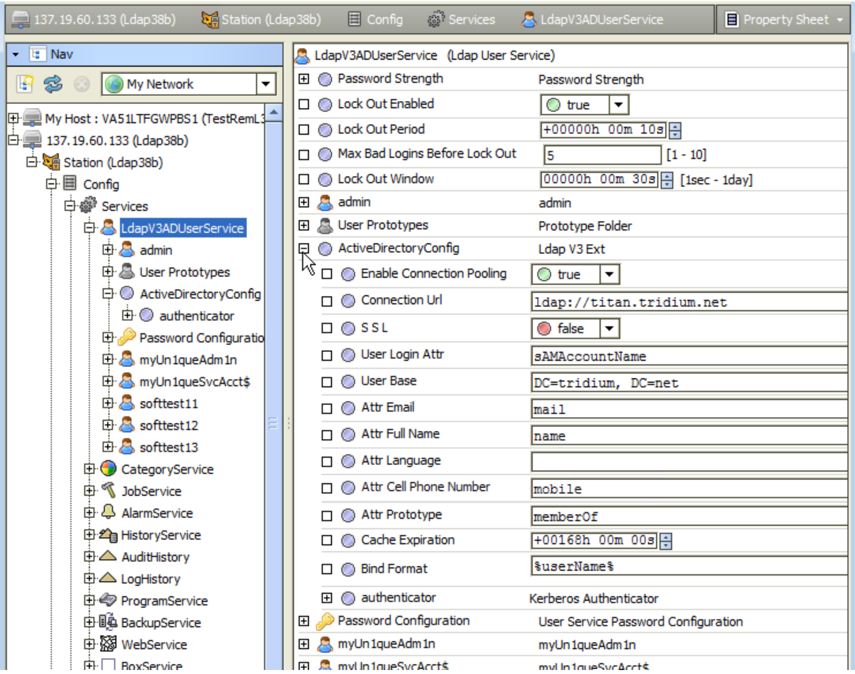
- **LdapV3ADUserService**
  (and children) The NiagaraAX station user service to support a Windows Active Directory (AD) service that uses LDAP version 3 (LDAPv3), as well one using LDAPv2. The child ActiveDirectoryConfig component holds all properties needed to configure connection to the AD domain controller, including an "authenticator" container for properties used to authenticate to that AD domain controller.
  By default, Kerberos authentication is used, applicable to LDAPv3 systems configured to use Kerberos authentication. If needed, after copying this user service in the station you can *replace* the default authenticator with the "SimpleAuthenticator" copied from the `ldap` palette. For reference details, see "LdapV3ADUserService" on page 2-11.
- **Authenticators**
  (Apply to LdapV3UserService and LdapV3ADUserService only) Seperately available authentication containers for these two user services. The KerberosAuthenicator is used as default "authenticator".
  - **SimpleAuthenticator**
    Use this to *replace* the standard (Kerberos) "authenticator" in either LdapV3 user service if the system is *not configured for Kerberos*, or if the station is on a JACE-2/4/5 series platform, or it is for an LDAPv2 system. Properties for a connection user and password are used, and authentication mechanism choices are none, simple, CRAM-MD5, and DIGEST-MD5. For quick start details, see "Configuring the SimpleAuthenticator component" on page 1-11.
  - **KerberosAuthenticator**
    Identical to the default "authenticator" in either LdapV3 user service as copied from the palette. Properties include the Kerberos realm name, key distribution center URL, station (service) name previously defined, and path to locally-stored keytab file or password for the service. Kerberos is a "ticket" based client-server "mutual authentication" method using symmetric key cryptography. NiagaraAX station host configuration is more involved, and system users must also configure PCs (Workbench and/or client web browsers) with Kerberos-specific settings. For quick start details, see "Configuring the Kerberos Authenticator component" on page 1-12.
    Note: *Kerberos is not supported on a "J9 Java VM" (JACE-2/4/5) series platform. On a station running on such a platform, use the SimpleAuthenticator in place of the KerberosAuthenticator for either LdapV3 user service.*

All the different user services in the ldap module have the same default view: **`Ldap User Manager`**. This view is virtually identical to the **`User Manager`** view for the standard (baja) UserService, and functions in the same way for all *local* users. See "Ldap User Manager view difference" on page 2-3.

LdapV3ADUserService

Available starting in AX-3.8, the LdapV3ADUserService supports LDAPv3 client access of Windows Active Directory (AD) server, and is also backwards-compatible with an LDAPv2 system. Authentication to the AD server can use either Kerberos (if LDAPv3) or else some other SASL (Simple Authentication and Security Layer) mechanism such as CRAM-MD5, DIGEST-MD5, or simple authentication.

Figure 2-13 LdapV3ADUserService example, replacing standard UserService



137.19.60.133 (Ldap38b) | Station (Ldap38b) | Config | Services | LdapV3ADUserService | Property Sheet

Nav
My Network
- My Host : VA51LTFGWPBS1 (TestRemL3
- 137.19.60.133 (Ldap38b)
  - Station (Ldap38b)
    - Config
      - Services
        - LdapV3ADUserService
          - admin
          - User Prototypes
          - ActiveDirectoryConfig
            - authenticator
          - Password Configuratio
          - myUn1queAdm1n
          - myUn1queSvcAcct$
          - softtest11
          - softtest12
          - softtest13
        - CategoryService
        - JobService
        - AlarmService
        - HistoryService
        - AuditHistory
        - LogHistory
        - ProgramService
        - BackupService
        - WebService
        - BoxService

LdapV3ADUserService (Ldap User Service)

⊞ Password Strength	Password Strength
□ Lock Out Enabled	true
□ Lock Out Period	+00000h 00m 10s
□ Max Bad Logins Before Lock Out	5 [1 - 10]
□ Lock Out Window	00000h 00m 30s [1sec - 1day]
⊞ admin	admin
⊞ User Prototypes	Prototype Folder
⊟ ActiveDirectoryConfig	Ldap V3 Ext
□ Enable Connection Pooling	true
□ Connection Url	ldap://titan.tridium.net
□ SSL	false
□ User Login Attr	sAMAccountName
□ User Base	DC=tridium, DC=net
□ Attr Email	mail
□ Attr Full Name	name
□ Attr Language	
□ Attr Cell Phone Number	mobile
□ Attr Prototype	memberOf
□ Cache Expiration	+00168h 00m 00s
□ Bind Format	%userName%
⊞ authenticator	Kerberos Authenticator
⊞ Password Configuration	User Service Password Configuration
⊞ myUn1queAdm1n	myUn1queAdm1n
⊞ myUn1queSvcAcct$	myUn1queSvcAcct$

Due to the popularity of Windows Server products with LDAPv3 capability, this may be the most frequently used of the NiagaraAX LDAP user services. Most key configuration properties are in its **`ActiveDirectoryConfig`** container, shown expanded in Figure 2-13 above. See "ActiveDirectory-Config (V3)" for more details.

ActiveDirectoryConfig (V3)

Properties in the ActiveDirectoryConfig container are described as follows:

- **Enable Connection Pooling**
  Either 'true' (default) or 'false'. When 'true', connections are allowed to be shared and re-used. This can improve performance.
- **Connection URL**
  URL of your LDAP (Active Directory) server, usually in the form: `ldap://your.domain.net`
  If the server uses a "non-standard" port, include in the URL, e.g. `ldap://your.domain.net:999` (note standard LDAP ports are 389, or else 636 if SSL).
  Note the scheme `ldaps://your.domain.net` is not supported in Connection URL.
- **SSL**
  Either 'false' (default) or 'true'. If 'true', the *station* uses SSL to communicate with the LDAP server.
  Note: *If true, be sure to enable SSL in the station's NiagaraNetwork's FoxService (for Workbench-to-station access) and also station's WebService (for browser-to-station access).*
- **User Login Attr**
  The specific attribute in the LDAP directory for the desired user login name.
  Note: *For Active Directory, the default* `sAMAccountName` *value is always used.*
- **User Base**
  Sub-tree of the LDAP server in which users who can access this station can be found. At the very least, it must contain the domain components of the server's domain, e.g. `DC=`*`domain`*`, DC=net`
- **Attr Email**
  The specific attribute in the LDAP directory to store user's email address, the value of which populates the Niagara user's Email property. The Active Directory default value is: `email`
- **Attr Full Name**
  The specific attribute in the LDAP directory to store user's full name, the value of which populates the Niagara user's Full Name property. The Active Directory default value is: `name`
- **Attr Language**
  The specific attribute in the LDAP directory to store user's language, the value of which populates the Niagara user's Language property. There is no default value (is blank).

- **Attr Cell Phone Number**
  The specific attribute in the LDAP directory to store user's cell phone number, the value of which populates the Niagara user's Cell Phone Number property. The default value is: `mobile`
- **Attr Prototype**
  The specific attribute in the LDAP directory to use for mapping a *User Prototype* (under the user service's **`UserPrototypes`** container) to users. The Active Directory default value is: `memberOf`
  This mechanism uses an "attribute value"-to-"component name" *matching* method of selection, where if no "name-matching" User Prototype is found, the frozen DefaultPrototype is used (when making the User component for the LDAP user, upon initial station login).
  For related details, see "Configure User Prototypes" on page 1-6.
- **Cache Expiration**
  Specifies how long a user's password is effective in the station, before being set to expired. Set to some differential time in the future, this is used in case the LDAP server is unavailable, so that users can still login with active credentials. Note if using Kerberos authentication, this feature is still applicable even though the station never receives user passwords. Instead, the station verifies the corresponding Kerberos user ticket and uses the cached user.
- **Bind Format**
  (ActiveDirectoryConfig under LdapV3ADUserService only) If *not* using Kerberos, but instead the SimpleAuthenticator, it may be necessary to specify the *exact format* of the login name to send to the LDAP server. This can differ according to the LDAP server, and may be required more often when the "Authentication Choice" in the SimpleAuthenticator is DIGEST MD5. In some cases, just the user base and login name may be sufficient to find a user in the LDAP directory.
  Bind Format processes "BFormat" (Baja Format) syntax, with a default value of `%userName%`.
  This default value may handle most cases, especially with an ActiveDirectory. However, if in the SimpleAuthenticator you choose DIGEST MD5 for authentication, this may need to be changed. For example, in one Active Directory (AD) 2000, a change was necessary to: `username@`*`domain.com`*
  In another (non-Active Directory) example using an OpenLDAP server, this property under the LdapConfig container of an LdapV3UserService (using SimpleAuthenticator and DIGEST MD5) required being set to: `%userLoginAttr%=%userName%,%userBase%`
  Note: *For server-specific details, consult with the onsite LDAP administrator for assistance if this property value needs to be changed.*
- **authenticator**
  Container for properties that define how the station authenticates with the AD server, with choices being either the default KerberosAuthenticator or a SimpleAuthenticator. See "Configure the LDAP authenticator (LdapV3 only)" on page 1-11.

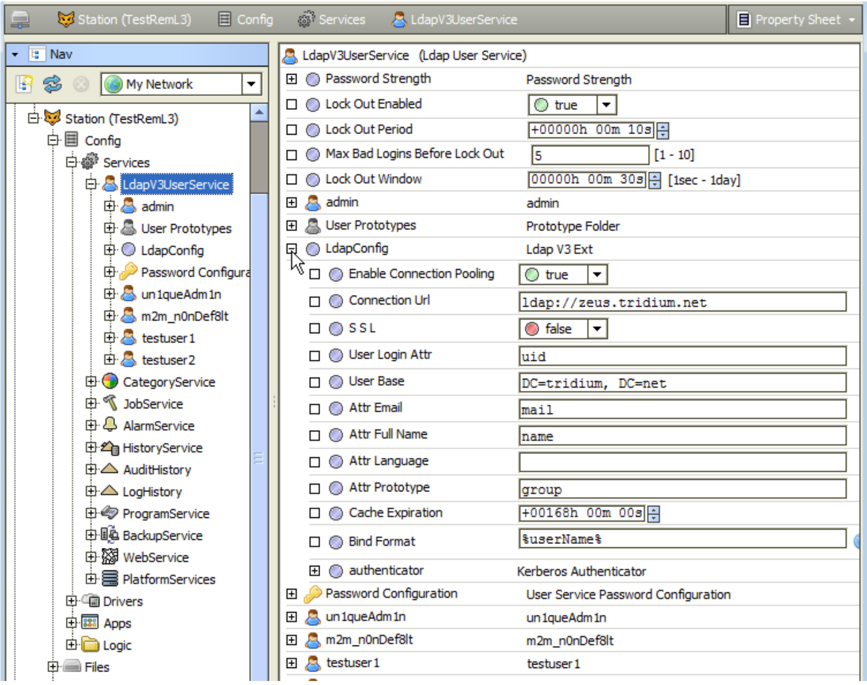
For related details, see:

- "Add and configure the LDAP user service" on page 1-3
- "Configure User Prototypes" on page 1-6
- "Configuring the ActiveDirectoryConfig component" on page 1-8

LdapV3UserService

Available starting in AX-3.8, the LdapV3UserService supports LDAPv3 client access of an LDAP server, for example OpenLDAP or OpenDS. Authentication to the LDAP server can use either Kerberos or else some other SASL (Simple Authentication and Security Layer) mechanism such as CRAM-MD5, DIGEST-MD5, or simple authentication.

Figure 2-14 LdapV3UserService example, replacing standard UserService



LdapV3UserService (Ldap User Service)

⊞ Password Strength	Password Strength
□ Lock Out Enabled	true
□ Lock Out Period	+00000h 00m 10s
□ Max Bad Logins Before Lock Out	5 [1 - 10]
□ Lock Out Window	00000h 00m 30s [1sec - 1day]
⊞ admin	admin
⊞ User Prototypes	Prototype Folder
⊟ LdapConfig	Ldap V3 Ext
□ Enable Connection Pooling	true
□ Connection Url	ldap://zeus.tridium.net
□ SSL	false
□ User Login Attr	uid
□ User Base	DC=tridium, DC=net
□ Attr Email	mail
□ Attr Full Name	name
□ Attr Language	
□ Attr Prototype	group
□ Cache Expiration	+00168h 00m 00s
□ Bind Format	%userName%
⊞ authenticator	Kerberos Authenticator
⊞ Password Configuration	User Service Password Configuration
⊞ un1queAdm1n	un1queAdm1n
⊞ m2m_n0nDef8lt	m2m_n0nDef8lt
⊞ testuser1	testuser1

Most key configuration properties are in its **`LdapConfig`** container, shown expanded in Figure 2-14 above. See "LdapConfig (V3)" for more details.

LdapConfig (V3)

Properties in the LcapConfig container are described as follows:

- **Enable Connection Pooling**
  Either 'true' (default) or 'false'. When 'true', connections are allowed to be shared and re-used. This can improve performance.
- **Connection URL**
  URL of your LDAP server, usually in the form: `ldap://your.domain.net`
  If the server uses a "non-standard" port, include in the URL, e.g. `ldap://your.domain.net:999` (note standard LDAP ports are 389, or else 636 if SSL).
  Note the scheme `ldaps://your.domain.net` is not supported in Connection URL.
- **SSL**
  Either 'false' (default) or 'true'. If 'true', the *station* uses SSL to communicate with the LDAP server.
  Note: *If true, be sure to enable SSL in the station's NiagaraNetwork's FoxService (for Workbench-to-station access) and also station's WebService (for browser-to-station access).*
- **User Login Attr**
  The specific attribute in the LDAP directory for the desired user login name.
  Note: *Different LDAP servers use a different attribute. For OpenLDAP, the attribute is:* `uid`
- **User Base**
  Sub-tree of the LDAP server in which users who can access this station can be found. At the very least, it must contain the domain components of the server's domain, e.g. `DC=`*`domain`*`, DC=net`
- **Attr Email**
  The specific attribute in the LDAP directory to store user's email address, the value of which populates the Niagara user's Email property. There is no default value (is blank).
- **Attr Full Name**
  The specific attribute in the LDAP directory to store user's full name, the value of which populates the Niagara user's Full Name property. There is no default value (is blank).
- **Attr Language**
  The specific attribute in the LDAP directory to store user's language, the value of which populates the Niagara user's Language property. There is no default value (is blank).
- **Attr Cell Phone Number**
  The specific attribute in the LDAP directory to store user's cell phone number, the value of which populates the Niagara user's Cell Phone Number property. There is no default value (is blank).

- **Attr Prototype**
  The specific attribute in the LDAP directory to use for mapping a *User Prototype* (under the user service's **`UserPrototypes`** container) to users. There is no default value (is blank).
  This mechanism uses an "attribute value"-to-"component name" *matching* method of selection, where if no "name-matching" User Prototype is found, the frozen DefaultPrototype is used (when making the User component for the LDAP user, upon initial station login).
  For related details, see "Configure User Prototypes" on page 1-6.
- **Cache Expiration**
  Specifies how long a user's password is effective in the station, before being set to expired. Set to some differential time in the future, this is used in case the LDAP server is unavailable, so that users can still login with active credentials. Note if using Kerberos authentication, this feature is still applicable even though the station never receives user passwords. Instead, the station verifies the corresponding Kerberos user ticket and uses the cached user.
- **Bind Format**
  (LdapConfig under LdapV3UserService only) If *not* using Kerberos, but instead the SimpleAuthenticator, it may be necessary to specify the *exact format* of the login name to send to the LDAP server. This can differ according to the LDAP server, and may be required more often when the "Authentication Choice" in the SimpleAuthenticator is DIGEST MD5. In some cases, just the user base and login name may be sufficient to find a user in the LDAP directory.
  Bind Format processes "BFormat" (Baja Format) syntax, with a default value of `%userName%`.
  This default value may handle most cases. However, if in the SimpleAuthenticator you choose DIGEST MD5 for authentication, this may need to be changed. For example using an OpenLDAP server, this property under the LdapConfig container of an LdapV3UserService (using SimpleAuthenticator and DIGEST MD5) required being set to:
  `%userLoginAttr%=%userName%,%userBase%`
  Note: *For server-specific details, consult with the onsite LDAP administrator for assistance if this property value needs to be changed.*
- **authenticator**
  Container for properties that define how the station authenticates with the LDAP server, with choices being either the default KerberosAuthenticator or a SimpleAuthenticator. See "Configure the LDAP authenticator (LdapV3 only)" on page 1-11.

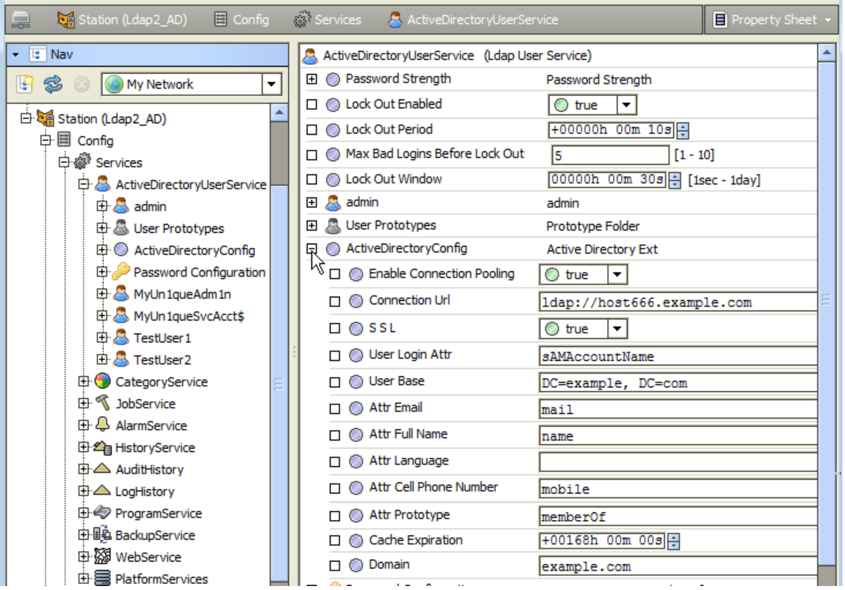
For related details, see:

- "Add and configure the LDAP user service" on page 1-3
- "Configure User Prototypes" on page 1-6
- "Configuring the LdapConfig component" on page 1-9

ActiveDirectoryUserService

The ActiveDirectoryService supports LDAPv2 client access of Windows Active Directory (AD) server. Connection to the AD server uses simple authentication.

Figure 2-15 *ActiveDirectoryUserService example, replacing standard UserService*



ActiveDirectoryUserService (Ldap User Service)

Password Strength	Password Strength
Lock Out Enabled	true
Lock Out Period	+00000h 00m 10s
Max Bad Logins Before Lock Out	5 [1 - 10]
Lock Out Window	00000h 00m 30s [1sec - 1day]
admin	admin
User Prototypes	Prototype Folder
ActiveDirectoryConfig	Active Directory Ext
Enable Connection Pooling	true
Connection Url	ldap://host666.example.com
SSL	true
User Login Attr	sAMAccountName
User Base	DC=example, DC=com
Attr Email	mail
Attr Full Name	name
Attr Language	
Attr Cell Phone Number	mobile
Attr Prototype	memberOf
Cache Expiration	+00168h 00m 00s
Domain	example.com

This is the same AD-specific LDAP user service that has been available since AX-3.1. Key configuration properties are in its **`ActiveDirectoryConfig`** container, shown expanded in Figure 2-13 above. See "ActiveDirectoryConfig (V2)" for more details.

ActiveDirectoryConfig (V2)

Properties in the ActiveDirectoryConfig container of the ActiveDirectoryUserService are described as follows:

- **Enable Connection Pooling**
  Either 'true' (default) or 'false'. When 'true', connections are allowed to be shared and re-used. This can improve performance.
- **Connection URL**
  URL of your LDAP (Active Directory) server, usually in the form: `ldap://your.domain.net`
  If the server uses a "non-standard" port, include in the URL, e.g. `ldap://your.domain.net:999` (note standard LDAP ports are 389, or else 636 if SSL).
  Note the scheme `ldaps://your.domain.net` is not supported in Connection URL.
- **SSL**
  Either 'false' (default) or 'true'. If 'true', the *station* uses SSL to communicate with the LDAP server.
  Note: *If true, be sure to enable SSL in the station's NiagaraNetwork's FoxService (for Workbench-to-station access) and also station's WebService (for browser-to-station access).*
- **User Login Attr**
  The specific attribute in the LDAP directory for the desired user login name.
  Note: *For Active Directory, the default* `sAMAccountName` *value is always used.*
- **User Base**
  Sub-tree of the LDAP server in which users who can access this station can be found. At the very least, it must contain the domain components of the server's domain, e.g. `DC=`*`domain`*`, DC=net`
- **Attr Email**
  The specific attribute in the LDAP directory to store user's email address, the value of which populates the Niagara user's Email property. The Active Directory default value is: `email`
- **Attr Full Name**
  The specific attribute in the LDAP directory to store user's full name, the value of which populates the Niagara user's Full Name property. The Active Directory default value is: `name`
- **Attr Language**
  The specific attribute in the LDAP directory to store user's language, the value of which populates the Niagara user's Language property. There is no default value (is blank).

- **Attr Cell Phone Number**
  The specific attribute in the LDAP directory to store user's cell phone number, the value of which populates the Niagara user's Cell Phone Number property. The default value is: `mobile`
- **Attr Prototype**
  The specific attribute in the LDAP directory to use for mapping a *User Prototype* (under the user service's **`UserPrototypes`** container) to users. The Active Directory default value is: `memberOf`
  This mechanism uses an "attribute value"-to-"component name" *matching* method of selection, where if no "name-matching" User Prototype is found, the frozen DefaultPrototype is used (when making the User component for the LDAP user, upon initial station login).
  For related details, see "Configure User Prototypes" on page 1-6.
- **Cache Expiration**
  Specifies how long a user's password is effective in the station, before being set to expired. Set to some differential time in the future, this is used in case the LDAP (AD) server is unavailable, so that users can still login with active credentials.
- **Domain**
  The value of this property is combined with the user's login name when authenticating against the server. For example, if the Domain property value is "`example.com`" and User Login Attr property value is "`sAMAccountName`", the ActiveDirectoryService would attempt to authenticate `janedoe` as `jandoe@example.com`.

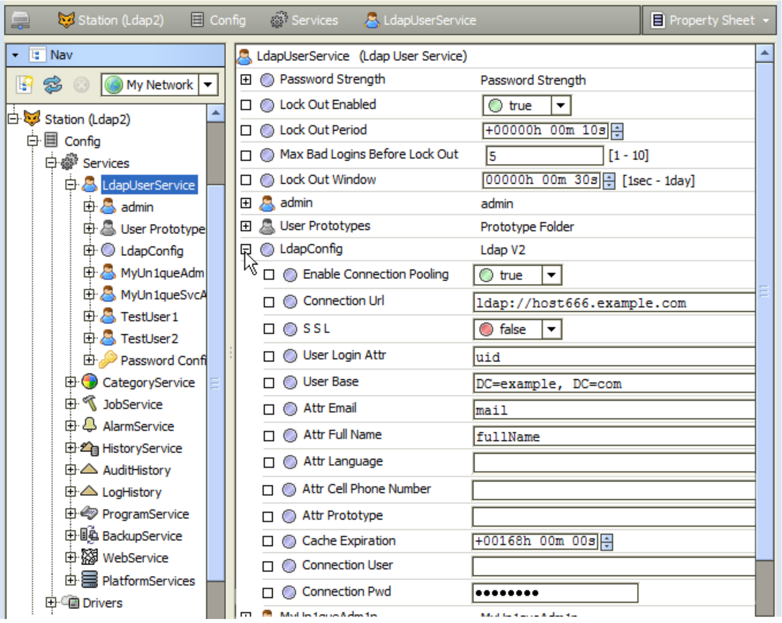
For related details, see:

- "Add and configure the LDAP user service" on page 1-3
- "Configure User Prototypes" on page 1-6
- "Configuring the ActiveDirectoryConfig component" on page 1-8

LdapUserService

The LdapUserService supports LDAPv2 client access of an LDAP server, for example OpenLDAP or OpenDS. Connection to the LDAP server uses simple authentication.

Figure 2-16 *LdapUserService example, replacing standard UserService*



LdapUserService (Ldap User Service)
⊞ Password Strength — Password Strength
□ Lock Out Enabled — true
□ Lock Out Period — +00000h 00m 10s
□ Max Bad Logins Before Lock Out — 5 [1 - 10]
□ Lock Out Window — 00000h 00m 30s [1sec - 1day]
⊞ admin — admin
⊞ User Prototypes — Prototype Folder
⊟ LdapConfig — Ldap V2
□ Enable Connection Pooling — true
□ Connection Url — ldap://host666.example.com
□ S S L — false
□ User Login Attr — uid
□ User Base — DC=example, DC=com
□ Attr Email — mail
□ Attr Full Name — fullName
□ Attr Language —
□ Attr Cell Phone Number —
□ Attr Prototype —
□ Cache Expiration — +00168h 00m 00s
□ Connection User —
□ Connection Pwd — ••••••••

This is the same generic LDAP user service that has been available since AX-3.1. Most key configuration properties are in its **`LdapConfig`** container, shown expanded in Figure 2-16 above. See "LdapConfig (V2)" for more details.

LdapConfig (V2)

Properties in the LdapConfig container of the LdapUserService are described as follows:

- **Enable Connection Pooling**
  Either 'true' (default) or 'false'. When 'true', connections are allowed to be shared and re-used. This can improve performance.

- **Connection URL**
  URL of your LDAP server, usually in the form: `ldap://your.domain.net`
  If the server uses a "non-standard" port, include in the URL, e.g. `ldap://your.domain.net:999` (note standard LDAP ports are 389, or else 636 if SSL).
  Note the scheme `ldaps://your.domain.net` is not supported in Connection URL.
- **SSL**
  Either 'false' (default) or 'true'. If 'true', the *station* uses SSL to communicate with the LDAP server.
  Note: *If true, be sure to enable SSL in the station's NiagaraNetwork's FoxService (for Workbench-to-station access) and also station's WebService (for browser-to-station access).*
- **User Login Attr**
  The specific attribute in the LDAP directory for the desired user login name.
  Note: *Different LDAP servers use a different attribute. For OpenLDAP, the attribute is:* `uid`
- **User Base**
  Sub-tree of the LDAP server in which users who can access this station can be found. At the very least, it must contain the domain components of the server's domain, e.g. `DC=`*`domain`*`, DC=net`
- **Attr Email**
  The specific attribute in the LDAP directory to store user's email address, the value of which populates the Niagara user's Email property. There is no default value (is blank).
- **Attr Full Name**
  The specific attribute in the LDAP directory to store user's full name, the value of which populates the Niagara user's Full Name property. There is no default value (is blank).
- **Attr Language**
  The specific attribute in the LDAP directory to store user's language, the value of which populates the Niagara user's Language property. There is no default value (is blank).
- **Attr Cell Phone Number**
  The specific attribute in the LDAP directory to store user's cell phone number, the value of which populates the Niagara user's Cell Phone Number property. There is no default value (is blank).
- **Attr Prototype**
  The specific attribute in the LDAP directory to use for mapping a *User Prototype* (under the user service's **`UserPrototypes`** container) to users. There is no default value (is blank).
  This mechanism uses an "attribute value"-to-"component name" *matching* method of selection, where if no "name-matching" User Prototype is found, the frozen DefaultPrototype is used (when making the User component for the LDAP user, upon initial station login).
  For related details, see "Configure User Prototypes" on page 1-6.
- **Cache Expiration**
  Specifies how long a user's password is effective in the station, before being set to expired. Set to some differential time in the future, this is used in case the LDAP server is unavailable, so that users can still login with active credentials.
- **Connection User**
  The user name for the initial LDAP server connection. It may be required if users who will be logging in are in different sub-trees of the LDAP directory. If the LDAP server supports anonymous connections, leave this property empty (blank).
- **Connection Pwd**
  The password for the user specified in property Connection User.

For related details, see:

- "Add and configure the LDAP user service" on page 1-3
- "Configure User Prototypes" on page 1-6
- "Configuring the LdapConfig component" on page 1-9

CHAPTER 3

Ldap component and plugin guides

These component and plugin guides provides summary help on components in the AX-3.8 `ldap` module, and also plugins (views) specific to any of those components.

- Ldap component reference summaries
- Ldap plugin reference summary

Ldap component reference summaries

Summary information is provided on components in the `ldap` module, listed alphabetically as follows:

- ActiveDirectoryExt
- KerberosAuthenticator
- LdapUserService
- LdapV2
- LdapV3Ext
- ProtoypeFolder
- SimpleAuthenticator

ldap-ActiveDirectoryExt

ActiveDirectoryExt (**`ActiveDirectoryConfig`**) is the frozen child "config" container under an **`ActiveDirectoryUserService`** (LDAPv2 support only) and contains properties that specify the connection URL, domain, and other attributes. For more details, see "Configuring the ActiveDirectoryConfig component" on page 1-8 and "ActiveDirectoryConfig (V2)" on page 2-16.

ldap-KerberosAuthenticator

The default "authenticator" in either LdapV3-compatible user service as copied from the `ldap` palette. Properties include the Kerberos realm name, key distribution center URL, station (service) name previously defined, and path to locally-stored keytab file or password for the service. Kerberos is a "ticket" based client-server "mutual authentication" method using symmetric key cryptography. NiagaraAX station host configuration is more involved than for simple authentication, and system users must also configure client PCs (Workbench and/or client web browsers) with Kerberos-specific settings.

Note: *Kerberos is not supported on a "J9 Java VM" (JACE-2/4/5) series platform. On a station running on this platform, use the SimpleAuthenticator in place of the KerberosAuthenticator.*

For related details, see the following:

- "Kerberos prerequisites" on page 1-2
- "Configuring the Kerberos Authenticator component" on page 1-12
- "Additional Kerberos client-side setup" on page 1-13
  - "Browser-independent setup" on page 1-14
  - "Browser-specific setup" on page 1-15
- "Kerberos and NiagaraAX login operation" on page 2-5
  - "Workbench access via Kerberos" on page 2-5
  - "Browser access via Kerberos" on page 2-7

ldap-LdapUserService

A station user service to support either an LDAP *version 2*-based LDAP server (if the ActiveDirectoryUserService or LdapUserService) or else an LDAP version 3-based LDAP server, if the LdapV3ADUserService or LdapV3UserService. (Note these last two are also backwards-compatible with an LDAPv2-based system.) In any case, the user service replaces the standard UserService in a station.

For more details, see the following:

- "LDAP / Active Directory Quick Start" on page 1-1
- "NiagaraAX LDAP Concepts and References" on page 2-1
- "LdapV3ADUserService" on page 2-11
- "LdapV3UserService" on page 2-13
- "LdapUserService" on page 2-17
- "ActiveDirectoryUserService" on page 2-16

ldap-LdapV2

LdapV2 (**LdapConfig**) is the child "LdapConfig" container under the **LdapUserService** (LDAPv2 support only) and contains various properties that define the LDAP attribute and connection methods.

For more details, see the following:

- "LdapConfig (V2)" on page 2-17
- "Configuring the LdapConfig component" on page 1-9

ldap-LdapV3Ext

LdapV3Ext is the frozen child "LdapConfig" or "ActiveDirectoryConfig" container under the **LdapV3ADUserService** or **LdapV3UserService** (LDAPv3 support) and contains various properties that specify the LDAP attribute and connection methods. In addition, a child "authenticator" container specifies authentication parameters for connection to the LDAP server.

Although intended for LDAPv3 support, these LDAP user services are also backwards-compatible with LDAPv2 systems. However, Kerberos authentication is not supported in this case.

For more details, see the following:

- "ActiveDirectoryConfig (V3)" on page 2-12
- "LdapConfig (V3)" on page 2-14
- "Configuring the ActiveDirectoryConfig component" on page 1-8
- "Configuring the LdapConfig component" on page 1-9

ldap-PrototypeFolder

PrototypeFolder (**User Prototypes**) is a frozen container under any of the Ldap user services (**ActiveDirectoryUserService**, **LdapUserService**, **LdapV3ADUserService**, **LdapV3UserService**). It contains a single frozen "Default Prototype" prototype user, as well as any additional prototype users. To create a new prototype, right-click "User Prototypes", and select the action "New Prototype".

For related details, see the following:

- "Configure User Prototypes" on page 1-6
- "Notes on User Prototypes in an LDAP user service" on page 1-7
- "Mapping of permissions and other preferences to LDAP users" on page 2-5

ldap-SimpleAuthenticator

Used to *replace* the standard (Kerberos) "authenticator" in either LdapV3 user service, necessary if the LDAP system is *not configured for Kerberos*, or if the station is running on a "J9 Java VM" platform (JACE-2/4/5 series), or it is for an LDAPv2-based system. Properties for a connection user and password are used, and authentication mechanism choices are none, simple, CRAM-MD5, and DIGEST-MD5.

For more details, see "Configuring the SimpleAuthenticator component" on page 1-11.

Ldap plugin reference summary

Summary information on plugins (views) unique to components in the `ldap` module are as follows:

- LdapUserManager

ldap-LdapUserManager

The **Ldap User Manager** is the default view on any of the Ldap or LdapV3 user service components (sourced from the `ldap` module). This view allows you to manage Ldap users as well as "local" station users. For example, you can add new local users and edit or delete existing local users. In the regard to local users, the Ldap User Manager works identically to the standard (baja) User Manager.

For more details, see "Ldap User Manager view difference" on page 2-3. For an overview, see "LDAP and NiagaraAX operation" on page 2-2.

Appendix A

Notes on Active Directory with Kerberos

In order to use Kerberos and Active Directory for station access via browsers, the IT (Kerberos) administrator must create service names for stations in the Kerberos database, and generate corresponding keytab files. This section provides a few notes on performing these tasks.

Note: *Only summary information is provided here, without conceptual details or consideration of possible variations in different versions of an Active Directory interface. Consult other resources about using Kerberos with Active Directory for detailed information.*

You may find additional details by searching the Internet on the following:

- Kerberos Interoperability Step-by-Step Guide for Windows Server 2003 *(Microsoft TechNet article)*

Other resources found may also be beneficial, depending on the Active Directory (Windows Server) version.

See the following appendix notes for more information:

- "Setting up a record in Active Directory for Kerberos Authentication"
- "Example setup notes from Tridium IT manager"

Setting up a record in Active Directory for Kerberos Authentication

A configuration overview for doing this in Windows 2000 and Active Directory is found in the MSDN (Microsoft Developer Network) article *HTTP-Based Cross-Platform Authentication by Using the Negotiate Protocol*, in section "Kerberos Infrastructure Configuration". Currently, this article section is at URL: http://msdn.microsoft.com/en-us/library/ms995329.aspx#http-sso-1_topic3

From this article, the high-level process is as follows:

1. Create a User Account in Active Directory for each NiagaraAX host (JACE) and service.
2. Create the SPNs associated with each User Account—this must be done on a Domain Controller.
3. Generate the keytab files for each service.
4. Authenticate the UNIX hosts to the Kerberos realm.
5. Copy the keytab files to the NiagaraAX hosts (JACEs).

Greater detail on the first three steps above are in the article's following section, "Commands for SPNs, account mappings and keytab files".

Commands for SPNs, account mappings and keytab files

Again, from the referenced MSDN article:

1. Create the user account in AD using the Active Directory Users and Computers Snap-in, and set the password. Do NOT select "User must change password at next logon" (remember the password). The account does NOT need to be a FQDN (fully qualified domain name)—just the host name. For example, if a JACE's FQDN is `myJace.example.com`, the account name should simply be "`myJace`".
2. Create the SPNs associated with this account on the KDC:

```
setspn–A host/mysrvr.example.com mysrvr
setspn–A HTTP/mysrvr.example.com mysrvr
```

Note: *Use upper-case "HTTP" to match the way Internet Explorer builds SPNs. Alternatively, you can run ktpass (as shown below) to create the SPNs. When using the "-princ" option of ktpass, you are specifying the Kerberos principal that is to be registered in the realm; the "-mapuser" option should be the account created in step 1 (above). Whether or not you run setspn, ktpass must be run to generate the necessary keytab files.*

3. Create and export the keytabs:

```
ktpass -princ host/mysrvr.example.com@EXAMPLE.COM -pass
<password> -mapuser mysrvr -out c:\temp\mysrvr.host.keytab
ktpass -princ HTTP/mysrvr.example.com@EXAMPLE.COM -pass
<password> -mapuser mysrvr -out c:\temp\mysrvr.HTTP.keytab
```

Example setup notes from Tridium IT manager

In configuring Tridium's own test setup for using Kerberos authentication with the AX-3.8 LDAP driver, the IT manager recorded these notes after following information in the referenced MSDN article ("Setting up a record in Active Directory for Kerberos Authentication" on page A-1).

1. Assigned a DHCP reservation to the Niagara host running the service. (Static IP would also work).
2. Created a DNS A record in `tridium.net` domain for this host (`kerbtest2`).
3. Created user account on `tridium.net` Active Directory (`kerbtest2`), and selected "Account is trusted for delegation" in the "Account Options" panel.
4. Created SPN:
   - **`setspn -A host/kerbtest2.tridium.net kerbtest2`**
   - **`setspn -A HTTP/kerbtest2.tridium.net kerbtest2`**
5. Created keytab files to be used on device running the service:
   - **`ktpass -princ HTTP/kerbtest2.tridium.net@TRIDIUM.NET -pass password -mapuser kerbtest2 -out c:\temp\kerbtest2.http.keytab`**
   - **`ktpass -princ host/kerbtest2.tridium.net@TRIDIUM.NET -pass password -mapuser kerbtest2 -out c:\temp\kerbtest2.http.keytab`**

The ktpass command created the files, but also generated this warning:

"`WARNING: pType and account do not match. This might cause problems.`"

However, the files worked, so we disregarded the warning.