Technical Document

Niagara LDAP Guide

December 23, 2024

niagara4

Legal Notice

Tridium, Incorporated

3951 Western Parkway, Suite 350
Richmond, Virginia 23233
U.S.A.

Confidentiality

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. 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. Tridium, JACE, Niagara Framework, and Sedona Framework are registered trademarks, and Workbench are trademarks of Tridium Inc. All other product names and services mentioned in this publication that are 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.

The product(s) described herein may be covered by one or more U.S. or foreign patents of Tridium.

For an important patent notice, please visit: http://www.honpat.com.

Contents

About this guide

This topic contains important information about the purpose, content, context, and intended audience for this document.

Product documentation

This document is part of the Niagara technical documentation library. Released versions of Niagara software include a complete collection of technical information that is provided in both online help and PDF format. The information in this document is written primarily for Systems Integrators. To make the most of the information in this book, readers should have some training or previous experience with Niagara software, as well as experience working with JACE network controllers.

Document content

LDAP (Lightweight Directory Access Protocol) manages user authentication using a database stored on a separate server. This guide introduces LDAP user authentication in the context of the Niagara Network.

Document change log

This topic summarizes changes and additions made to this document.

December 23, 2024

- Added new topic "Prevent Duplicate Usernames" in the "Introduction to LDAP" Chapter.

June 21, 2023

- Edited note in "Using a browser and Kerberos to log in with a single sign on" topic to add JACE-9000.

June 1, 2021

- Added "baja-UserPrototypeProperty" and "ldap-KerberosConfig" to the "Components and Plugins" chapter. Added a new task topic, "Setting up an alternate default prototype" to the "Setup and Configuration" chapter. Corrected id for ldap-LdapAuthenticationScheme.

October 10, 2018

- In the "Plugins and Components" chapter, added the topic "baja-UserPrototype."

August 9, 2017

- In the topic "baja-UserService," added the description about "Effect of property changes on user session."

Related documentation

This topic lists the other documents that relate to the information contained in this guide.

- User authentication, FoxService, and WebService are documented in the *Niagara Station Security Guide*.

Chapter 1. Setup and configuration

The **AuthenticationService** manages all LDAP authentication requests. The types of authentication a station supports, in this case LDAP, are determined solely by this service. The most important element in LDAP authentication is the LDAP authentication scheme. A station supports multiple schemes, with each user account tied to a specific scheme.

Use this checklist to guide the setup and configuration process:

- All prerequisites fulfilled.
- LDAP authentication scheme selected and properties configured.
- User prototypes set up. See.
- Local super user and service users set up.
- If you are using the **`KerberosScheme`**, PC client configured for Kerberos.

  Each host configured to access the Key Distribution Center.
- Connection using a browser confirmed.
- Browser configured for LDAP support.

Prerequisites

Before you can configure your hosts for LDAP authentication your stations need to be licensed, you need to collect information from your LDAP and Kerberos administrators, as well as provide information to your LDAP administrator.

Licensing

Each Niagara platform (Supervisor and JACE) must be licensed for LDAP user services.

- The LDAPv2-compatible authentication scheme does not require host licensing. This is effectively the same LDAP authentication scheme provided since Niagara 4. They do not offer Kerberos as an authentication choice.
- To use Kerberos authentication, your host platform must be licensed for LDAPv3. The following is an example of the license line:

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

LDAP environment and properties

Each Niagara host (Supervisor and controller) must be on a network with an existing LDAP server. The server must support LDAPv2 or later.

You need at least the following information from your LDAP system administrator:

- URL for the LDAP server (ldap://`<your.domain.net:nnn>` where `<your.domain.net:nnn>` is the URL for the LDAP server, and `<nnn>` is any port other than the standard, default LDAP port. To use a standard port (389, or 636 if you are using SSL/TLS), you do not need to include the port in the URL.
- User names for logging in to each station as they appear in the LDAP directory.

Information your LDAP system administrator may need from you

- The name of the user prototype (group) to associate with each user (such as, manager, operator, etc.).
- Your name for each station.

Kerberos prerequisites

You need the following information from your Kerberos administrator:

- Kerberos realm name (should be in UPPERCASE).

- Key Distribution Center URL.
- A service name (based on the station name you provided) for each station. This URL-style name must be set up by your Kerberos administrator on the LDAP server. This name should be in the form:

  http/<somename>.<domain.com>

  where <somename> is the name by which you will access your station via a browser, and <domain.com> is your realm.

  This name must be trusted for delegation. If you are not planning for Kerberos authentication via the browser, you can use a regular user name (not a service).
- A keytab file or a password for each service name (station). Services typically require a keytab file, whereas users typically use a password.

FAQs

Use these questions and answers to broaden your understanding of LDAP and Niagara.

Q: Can I use SSL/TLS with LDAP?

A: Yes, in fact, you should configure all platforms and stations for TLS (Transport Layer Security). Refer to the *Station Security Guide*.

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

A: No. the Niagara network-user feature is incompatible with LDAP (and no hybrid system is supported). All centralized user management is provided by the LDAP server. Local station users, which are unique to each station, are supported.

Q: Is Kerberos always associated with LDAP in Niagara?

A: Kerberos is an available authentication scheme for LDAPv3.

Q: Can a station support an older LDAPv2 level server or Active Directory using the newer LDAPv3–compatible LDAP schemes?

Yes. These schemes are backwards-compatible with LDAPv2-based systems. However, Kerberos authentication is not available.

Q: Can I configure my 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 this restriction, Kerberos cannot be used when running in FIPS mode, as the algorithms it requires are not supported by the FIPS cryptographic provider.

Setting up the authentication scheme

The LDAP scheme defines the properties that are unique to LDAP authentication.

Prerequisites:
You are working on a computer using a secure connection to the network. You have opened Workbench

Step 1. Open the ldap palette.

Step 2. Drag an LDAP scheme (LdapScheme or KerberosScheme) to the station's **Config** > **Services** > **AuthenticationService** > **AuthenticationSchemes** container.

Step 3. To open the scheme property sheet, double-click the scheme name.
The property sheet opens.

Step 4. If you are configuring the LdapScheme, select the configuration type.

The LdapScheme supports three separate sets of configuration properties identified by the scheme type: Active Directory, Ldap V2, and Ldap V3. While all types share the same basic properties (**`Enable connection Pooling`**, **`Connection URL`**, **`SSL`**, and the attributes (attr) properties), each includes one or more additional properties.

The KerberosScheme supports a single set of configuration properties that include some of the same properties used by the LdapScheme.

Step 5. For each attribute property, enter the mnemonic required by the LDAP directory.

The attribute properties correspond to the names of the attributes in the LDAP directory. For example, to populate the **`Full Name`** property, enter `Fname`. The following lists some of the mnemonics you may use. For a complete list, contact your LDAP administrator.

For this Property	enter this mnemonic in the property field.
User Login Attr	• For ActiveDirectory use `sAMAccountName`. • For OpenLDAP, use `uid`.
Attr Email	`Email`
Attr Full Name	`Fname`
Attr Language	`Preferred Language`
Attr Prototype	`Prototype`

The following is an example of the attribute properties returned from an LDAP server.

```
User: jdoe
uid: jdoe
Fname: John Doe
Position: Software Engineer
Address: 123 Fake Street
Email: jdoe@email.com
Preferred Language: German
Groups: Engineering
```

Step 6. Configure the other properties based on the type of scheme and click **Save**.

Setting up user prototypes

When a new LDAP user logs in to a station for the first time, the system creates a user account in the **UserService** and names it based on the user name portion of the person's login credentials as stored on the LDAP server. The system then populates the **`Attr`** (attribute) properties, such as **`Full Name`**, **`Email`**, and **`Language`**, directly from the LDAP server. It populates other properties, such as **`Permissions`**, from the local user prototype in the station. If no prototype is identified for the user, the system populates a new user's properties (all except password) using values defined in the **Default Prototype**. Assigning a user prototype is a way to group users who share the same permissions. Customizing the **Default Prototype** properties before you create users can simplify the creation process even in a non-network-user scenario.

Prerequisites:
The station is open in Workbench.

Step 1. To configure the **Default Prototype**, right-click the **UserService** in the Nav tree and click **Views** > **AX Property Sheet**.

Step 2. Expand the **User Prototypes** node and double-click the **Default Prototype** node.

Step 3. Make changes to the properties that apply to all system users, and click **Save**.

To ease the burden of making new users, consider changing these properties: **Expiration**, **Authentication Scheme Name** and **Prototype Name**.
When they log in, any new LDAP users inherit these values as the default properties, including permissions. And these values appear as the defaults when you create a new user. You can change them for a specific user at any time.

Step 4. To make a custom prototype, get a list of the **attrPrototype** names from your LDAP administrator.
The **attr prototype** property usually defines the group to which the user belongs.
For example, if you have user prototypes named "sysIntegrator" and "buildingManager", an LDAP user who is a member of the buildingManager group on the LDAP server inherits permissions from the buildingManager prototype.

Step 5. To make a custom prototype, right-click the **Default Prototype** in the Nav tree and click **Duplicate**.
The **Name** window opens with the default name of `defaultPrototype1`.

Step 6. Change this name to the same name for the user group (type of user) on the LDAP server, such as Manager, Operator, Engineer, etc. and click **OK**.

Step 7. Repeat duplicating the **Default Prototype** and configuring properties until you have set up a separate prototype for each user group.
LDAP users may belong to multiple groups on the LDAP server, but they can only be assigned one prototype. If an LDAP user belongs to multiple groups that match prototype names, the system defaults to the first prototype in the prototypes folder.
For example, if you have prototypes named "sysIntegrator" and "buildingManager", with "sysIntegrator" being first in the list, and an LDAP user who is a member of both groups on the LDAP server, the user inherits permissions from the "sysIntegrator" prototype.

Step 8. When you are finished, save the station by right-clicking the station **Config** node on the Nav tree and clicking **Actions** > **Save**.

Setting up an alternate default prototype

You may configure an alternate default user prototype using the **baja-UserPrototype** component.

Prerequisites:
You are working in Workbench connected to a Supervisor or remote station.

Step 1. Open the baja or ldap palette.

Step 2. Drag a **UserPrototype** component from either the baja or ldap palettes to the **User Prototypes** folder under the **UserService**.

Step 3. Give the **UserPrototype** a name that matches the desired **attrPrototype** value.
attrPrototype is a property on the **Kerberos Config AX Property Sheet**.
The component displays a list of user properties including **Full Name**, **Enabled**, **Expiration**, etc.
Each property contains two sub-properties: **Overridable** and **value**.

Step 4. Configure each user property and click **Save**.
You now have an alternate default user prototype.

Step 5. To replace the standard default user prototype with the one you just created, navigate to the **User Prototypes** folder under the **UserService** and set the **Alternate Default Prototype** property to the new default user prototype.

Setting up local users

A local user, such as the admin, or other super user, may log in to a station using the standard login. This type of login provides access to only the resources of the local station.

Prerequisites:

Workbench is open and you are connected to the station

Step 1. To open the platform, choose one of the following:
- If you are working on a remote host, open a platform connection to the host. If you are upgrading this remote host, use the **Station Copier** tool to install the modified station back to the host.
- If you are working on a locally running station, such as on a Supervisor, open a local platform connection and `Start` the station from the **Application Manager** view.

Step 2. Allow sufficient time for the station to restart.
If you are configuring a controller, a station transfer results in a controller reboot first.

Step 3. In Workbench, open a station connection to the host as the admin super user.

Step 4. Double-click the **UserService** container in the Nav tree.
The property sheet opens.

NOTE: The **`Authenticator`** properties apply to local users only. LDAP authentication properties, including user name and password, are configured in the LDAP server/system, and not in Niagara. The **`Prototype Name`** applies only to users supported by LDAP.

Step 5. Create a new super user to replace the admin user.

For this user you do not need to enter LDAP attribute mnemonics. A local user does not appear in the LDAP directory on the LDAP server.

Step 6. After creating the new super user, delete the admin user since it is no longer a frozen property.

NOTE: Do not delete the admin user before you create the new super user.

Step 7. Create a new local user as a super user. Assign a user name that easily identifies the host platform, and a strong password.

Step 8. Update the other stations in the network with the proper credentials under **Client Connection** > **NiagaraStation** to recognize the newly-configured station.

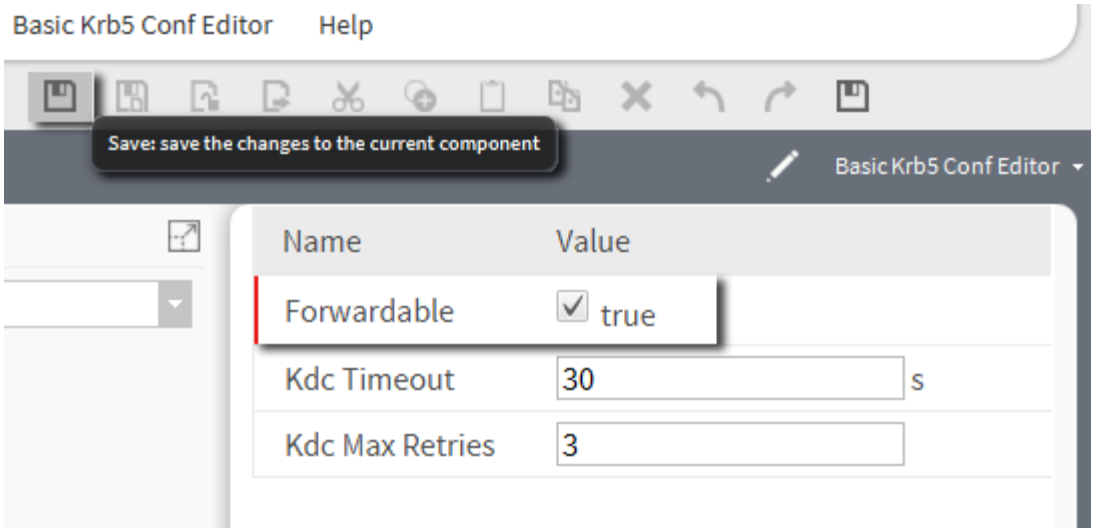
Result
Your local station now has two local users in addition to the now disabled admin user and the standard guest user.

Setting up a client PC for Kerberos

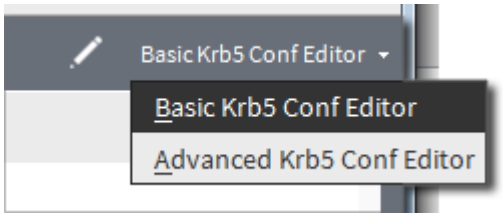
For any computer to access (as a client) a station that supports Kerberos authentication, you must update a Kerberos configuration file (`krb5`) in the PC with the default realm and define which flags to set on acquired tickets. (Kerberos authentication requires the ability to acquire Kerberos tickets that can be forwarded.) In addition, you must update the Windows registry.

Step 1. In Workbench, click **Tools** > **Kerberos Configuration Tool**.

Step 2. In the **Basic Krb5 Conf Editor** view, click the **`Forwardable`** checkbox to set the property value to “true” and click the toolbar icon to save your change, as shown here.



NOTE: If your Kerberos setup requires a more advanced `krb5.conf` configuration, you can manually configure the file using the **Advanced Krb5 Conf Editor** view, located under the **View** dropdown list, as shown here.



Also, if you are working with Linux, some systems may require a more advanced `krb5.conf` file. If that is the case, have your Kerberos administrator set-up this file for you.

Step 3. If your PC is running Windows XP SP2 or later, and you would like to access your native Kerberos ticket, you must set a registry key to allow Java to access the ticket.

a. Before setting a registry key, back up your Windows registry.

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

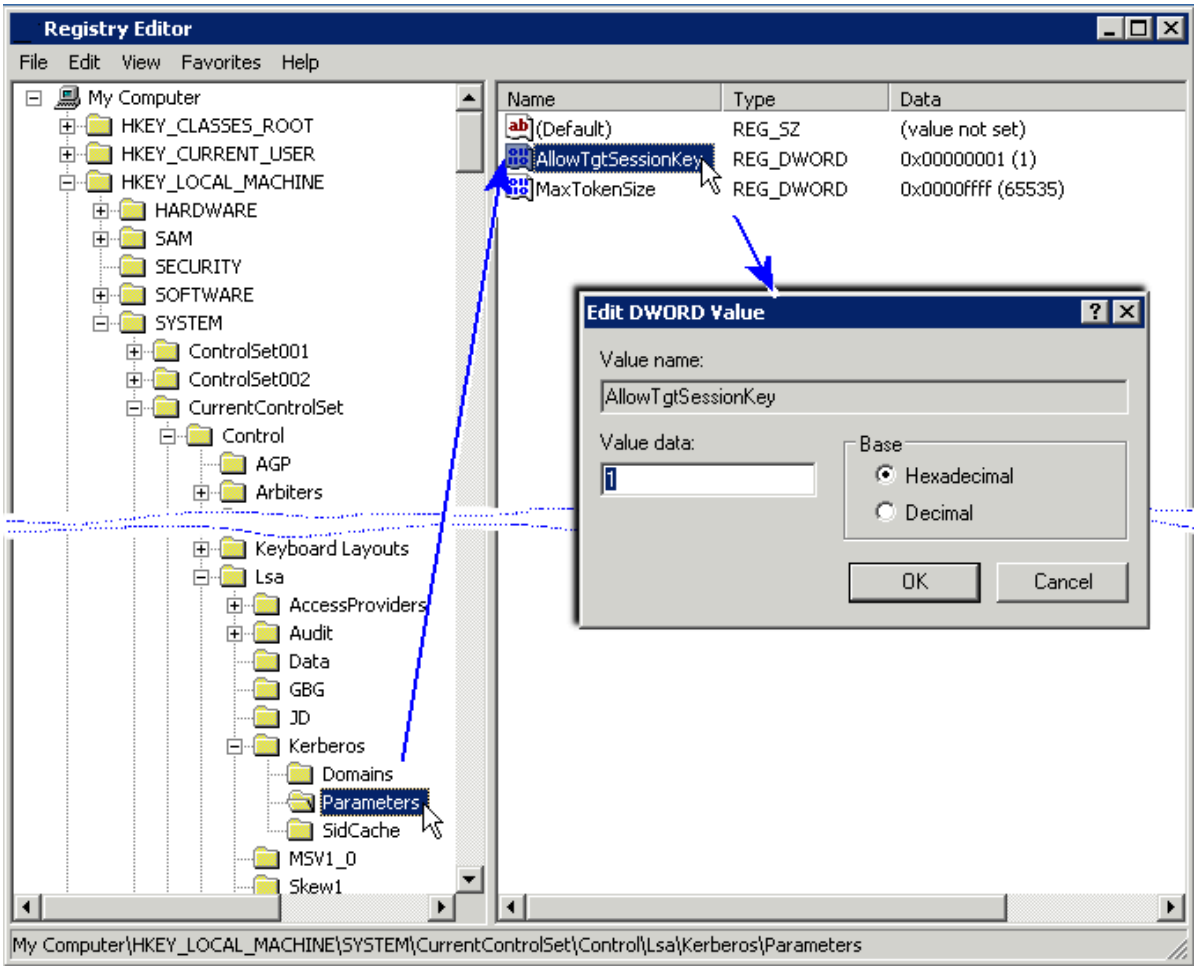
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters

Value name: AllowTgtSessionKeyValue type: REG_DWORDValue: 0x01

If configuring Windows XP, add or edit this key:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos

Value name: AllTgtSessionKeyValue type: REG_DWORDValue: 0x01



NOTE: If necessary, you can return to the default Windows security setting by changing the value of this registry key to zero (0).

Result

On completion of this procedure, you have successfully updated the Kerberos configuration file (`krb5.conf`) and set up a registery key in the PC.

Setting up access to the Key Distribution Center

Kerberos authentication issues authentication tickets, which the system uses in a similar manner to private-key authentication. Ticket processing involves retrieving a key from a KDC (Key Distribution Center). Kerberos uses reverse DNS (Domain Name System) to find the referenced Key Distribution Center. You must specify a reverse DNS entry for both the client and station DNS servers. Otherwise, users are unable to acquire Kerberos tickets and log in.

This procedure documents how to configure both a PC client and station to access a KDC. 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 and prone to error when dealing with multiple stations and multiple client machines. Setting up DNS servers with reverse DNS entries is the recommended best practice.

Step 1. Contact your IT administrator to see if the appropriate entry exists on the LDAP server.

If you do not have a workable reverse DNS entry, you may configure an entry in the `hosts` file on each client PC and station. This entry maps the IP address of the Key Distribution Center.

NOTE: Configuring mapping in the `hosts` file is acceptable for testing purposes, but is not recommended on a production system where the site is live and many people need to access it. It is important to note that having the proper DNS entries is far more desirable than modifying hosts files. If you find that the DNS entries do not already exist, request that your IT administrator add them.

On Windows PCs, the hosts file is located at `C:\Windows\System32\drivers\etc\hosts`.

On Linux `hosts` it is located at: `/etc/hosts`.

Step 2. Add the following entry in your client `hosts` file:
`<nnn.nnn.nnn.nnn> <kdc.domain.net>`

where `<nnn.nnn.nnn.nnn>` is the IP address of the KDC and `<kdc.domain.net>` is the domain name.

Step 3. On each platform, use the platform **TCP/IP Configuration** view (or equivalent view on the station's TcpIpPlatformService) to access and edit the `hosts` file with the same entry.

Making sure you can connect using a browser

Kerberos processes names and not IP addresses. The IP address of your PC must map to the name of the service you intend to use.

Step 1. Attempt to connect to the station using a fully-qualified domain name:
`http://<some.domain.com/somepage>`, where
`<some.domain.com/somepage>` is the LDAP server's domain name and home page name.

Step 2. If you are unable to connect, edit your client PC's `hosts` file to add an entry similar to:
`<nnn.nnn.nnn.nnn> <some.domain.com>`,where `<nnn.nnn.nnn.nnn>` is the IP address of the LDAP server,and `<some.domain.com>` is the domain name of the server.
For example,`<172.16.10.10> <kerbtest2.mydomain.net>`

This Ip address maps to the Kerberos service associated with kerbtest2 on mydomain.net.

Configuring Firefox

Using Firefox for browser access (as a Kerberos authenticated LDAP user) to a station requires that you add to the browser's security configuration the stations to which you wish the browser to connect .

NOTE: The following instructions are subject to change due to browser updates. Refer to the browsers' documentation for the latest instructions.

Step 1. Open a Firefox window.

Step 2. Type `about:config` in the location bar and press **Enter**.
If a warning appears, continue (promise to be careful).

Step 3. In the **Search** box near the top of the page type `negotiate`.
This filters the Firefox configuration attributes to six or seven. You need to edit these entries:

`network.negotiate-auth.delegation-uris`

`network.negotiate-auth.trusted-uris`

Step 4. Include the URLs of the station(s) that the browser needs to be able to access. Use a comma to separate multiple stations.
For example, if two stations have the following URLs: `http://host1.domain.com/somepage`, and `http:/ /host2.domain.com/somepage`, enter the URLs as follows:

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

Result
Firefox is ready for Kerberos authentication and you should be able to log in to stations without being prompted for a user name and password.

Configuring Internet Explorer

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

NOTE: The following instructions are subject to change due to browser updates. Refer to the browsers' documentation for the latest instructions.

Step 1. Open an Internet Explorer window.

Step 2. Using the menu bar, click **Tools** > **Internet Options**.
The **Internet Options** window opens.

Step 3. Click the **Security** tab and select the **Local intranet** zone.

Step 4. Click **Sites** > **Advanced**.
The **Add a website to this zone** window opens.

Step 5. Type in the URL for a station and click **Add**.
`http://<host1.domain.com>`
where `<host1.domain.com>` is the station's URL.
If you have multiple stations to add, continue typing in URLs and clicking **Add**.

Step 6. To return to the **Security** tab, click **Close** > **OK**.

Step 7. With the **Local intranet** zone selected, click the **Custom level...** button.
The **Security Settings — Local intranet** window opens.

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

Step 9. To close **Internet Options**, click **OK** twice.

Result
Internet Explorer should now be ready for Kerberos authentication and you should be able to log in to stations without being prompted for a user name and password.

Configuring Chrome security

Using Google Chrome for browser access (as a Kerberos authenticated LDAP user) to stations requires some client side setup.

NOTE: If you previously configured Internet Explorer to use Kerberos for LDAP access to stations, this may already be done. However, the Chrome startup arguments still need configuration.

Also, note that the following instructions are subject to change due to browser updates. Refer to the browsers' documentation for the latest instructions.

Step 1. Open a Google Chrome window.

Step 2. Click **Customize** > **Settings** or type `chrome:settings` in the location bar and press **Enter**.
The **Chrome Settings** page opens.

Step 3. Near the bottom of the page, click **Show advanced settings...**, scroll down to the **Network** section and click the **Change proxy settings...** button.
The **Internet Options** window opens.

Step 4. Click the **Security** tab, select the **Local intranet** zone, click **Sites** > **Advanced**.

The **Add website to this zone** window opens.

Step 5. Type in the URL for a station and click **Add**.
`http://host1.domain.com`
If you have multiple stations to add, continue typing in URLs and clicking **Add**.

Step 6. To return to the **Security** tab, click **Close** > **OK**.

Step 7. With the **Local intranet** zone selected, click the **Custom level...** button.
The **Security Settings — Local intranet** window opens.

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

Step 9. To close **Internet Options**, click **OK** twice.

Step10. Close all Chrome windows.

Configuring Chrome registry keys

In order for Kerberos authentication (without a logon prompt) to work correctly, you must configure a registry key in the client PC to hold the Kerberos delegation server whitelist for Chrome. This whitelist is the list of host names of each Niagara station you intend to authenticate to.

Step 1. To add the key, first start the Registry Editor (**Start** > **Run**... `regedit`).

CAUTION: Incorrectly editing the registry may damage your system. Before making changes to the registry, you should back up any valued data on your computer. Refer to the Registry Editor Help for complete details on configuring the registry.

Step 2. Expand the `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google` directories and create a new Key named: " `Chrome` ".

Step 3. Edit the newly created Chrome key to add a new **`String Value`** named:
" `AuthNegotiateDelegateWhitelist` ".

Step 4. Edit the new string value to configure its **`Value data`** field with a comma separated list (without quotes) of the host names of each station you would like to authenticate to.
For example: " `host1.domain.com,host2.domain.com` ".

Step 5. Close the registry window.

Result
Google Chrome is now configured for Kerberos authentication. You should be able to log in to stations without being prompted for a user name and password.

Configuring a Kerberos master-slave server

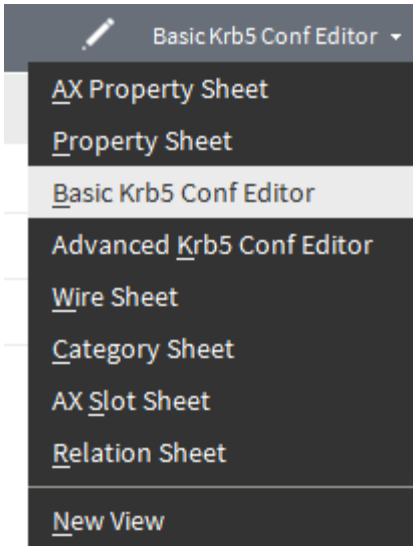
Many Kerberos/LDAP systems have redundant Kerberos/LDAP servers to provide load balancing and high availability. Typically, there will be one DNS entry that will resolve to each of the Kerberos/LDAP servers. For example, `example.com` may resolve to `dc1.example.com` and `dc2.example.com`. If the client fails to connect to the first entry, it will fail over to the next one. There are a few extra steps necessary to configure master-slave fail-over in Niagara.

Step 1. In the Kerberos Authentication Scheme, set your connection URL to one that will resolve to each of your LDAP servers (`ldap://example.com` in our example above).

Step 2. Set the Connection Timeout property to a reasonable time for your scenario.

Step 3. Set the Key Distribution Center to a hostname that will resolve to each of your key distribution centers (e.g. as`example.com` in our example above).

Step 4. Open the Basic Krb5 Conf Editor view on the Kerberos Authentication Scheme.



Step 5. Select and enter values for the Kdc Timeout and Kdc Max Retries properties.

Step 6. For any Workbench client that will authenticate to the station with Kerberos, navigate to **Tools** > **Kerberos Configuration Tool** and set the Kdc Timeout and Kdc Max Retries properties to the same values that you configured for the station, and set the Forwardable property to true.

Chapter 2. Introduction to LDAP

LDAP (Lightweight Directory Access Protocol) uses a separate server to provide an IP-network-accessible, hierarchical, and distributed database for storing information about authorized system users and their access privileges. Many network hosts can use the LDAP services, which are administered from a central location.

LDAP implementations

LDAP is an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services for an IP network. A common usage of LDAP is to provide a single sign on where a single user logs in to multiple network services using but one password.

Niagara supports two LDAP server implementations:

- Windows AD (Active Directory)

  This widely-implemented type of LDAP server is a Microsoft-supplied service used on Windows domain networks, and is included in most Windows Server operating systems. AD provides an interface for these protocols: LDAP (LDAPv2 or LDAPv3) and Kerberos (for authentication). With AD, users can access resources anywhere on the network with a single login.

  The Windows AD is structured as a hierarchical tree of objects.

  To integrate a Windows AD system with a network of Niagara stations, in the **Services** container and under the **AuthenticationService**, add one of the authentication schemes:

  - LdapScheme is for ADs versions LDAPv2, and LDAPv3.
  - KerberosScheme is for ADs that support Kerberos authentication. The host Niagara platform must be licensed for LDAPv3. If Kerberos authentication is used, the LDAPv3 requires the attribute: `Kerberos="true"`.
- Open source implementations

  These implementations, including Apache Directory Server and OpenLDAP, support both LDAPv2 and LDAPv3 (with the possibility of Kerberos authentication).

  Each of these implementations is structured as a hierarchical tree of objects. Each object has a set of attributes.

How LDAP benefits Niagara

LDAP communicates record-based, directory-like data between programs. It defines database access permissions and provides a schema, which is a way to describe the format and attributes of data stored in a server.

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

- Ease of implementation. Installations that already use Windows AD or an open-source implementation of LDAP can easily include stations in their existing user management configuration.
- Automatic new user account creation. When a user logs in to a station for the first time, the system automatically creates a user account (component) in the station and populates it with pre-defined properties (based on user prototype), such as permissions, and predefined LDAP properties (from the LDAP server), such as email address, full name, and language.
- Security. Kerberos authentication (available for LDAPv3-based AD or open source systems) offers a high level of security. Implementing Kerberos requires client setup of hosts and browsers.
- Simplified login. Current users may log in without needing to enter credentials.

NOTE: All stations on the network (both Supervisors and controllers) must use the LDAP server. The system does not support a mixture of stations using the standard UserService with other stations using an LDAP user service.

Local vs LDAP users

Once an LDAP authentication scheme is configured and running, most user access to a station comes from LDAP users. However, most configurations benefit from at least two regular station users that are not dependent upon LDAP server communications.

The two local users are:

- A replacement user for the admin user. The name "admin" is commonly used and easy for hackers to guess. Creating a new local super user with a unique name and strong password is a simple way to improve overall system security.
- A local service user you can reference in other remote stations when configuring the **`Client Connection`** properties under the remote station's **NiagaraStation** device.

  In theory, an LDAP user could serve as a service user, however, this is not recommended. A local service user makes the initial configuration of a **NiagaraNetwork** more straightforward and 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.

  NOTE: Do not allow any person to log in to the station using this user account. A service user is only for Fox station-to-station communications.

Configuration properties and LDAP user attributes

An LDAP server maintains a directory of information about system users. Each entry (record) in an LDAP directory consists of multiple attributes, which may or may not be assigned values. Users within the **NiagaraNetwork** require additional properties, such as permissions and facets that apply only within the **NiagaraNetwork** context.

A sample LDAP user entry might contain the following information from the LDAP server:

Figure 1. Example LDAP directory record

```
User: jdoe
uid: jdoe
Fname: John Doe
Position: Software Engineer
Address: 123 Fake Street
Email: jdoe@email.com
Preferred Language: German
Groups: Engineering
```

Several key configuration properties in each of the LDAP authentication schemes correspond directly to the names of attributes in the LDAP directory.

The property names for these LDAP properties begin with **`Attr`** (attribute). The system pulls the values for these properties from the LDAP directory on the LDAP server and uses them to fill out information about the user.

In the example above, the station user is `jdoe`. To populate the **`Full Name`** property value, you enter `displayName` in the **`Attr Full Name`** field.

The user properties that are not maintained by the LDAP server appear in the **UserService** property sheet for each user.

Automatic new user creation

All users must exist in the LDAP directory on the LADP server. When a new employee joins your team, make sure you set them up in the LDAP server before they attempt to log in to a station. An appropriate user prototype that contains default properties for each type of user should exist in each station. (User prototypes allow you to group users, for example: manager, operator, engineer, etc.).

When a new user logs in to a station for the first time, the system automatically creates a new user account (component) in the station. It uses the user name that the person logged in with (the person's user login name on the LDAP server) as the account name. The system populates (maps) this component's properties from two sources:

- It populates the properties the attr properties with attributes supplied from the LDAP server.

  One of those attributes identifies the group within your organization to which the user belongs. This attribute is the **`Attr Prototype`**.
- Niagara uses the **`Attr Prototype`** name to identify the user prototype in the station from which to populate the component's local user properties, including user permissions, facets, Nav file, default Web and Mobile profiles, and other specific properties required by a station.

  For Active Directory, this is the `memberOf` attribute.

  NOTE: For the automatic populating of Niagara user properties to work, the name of the **`Attr Prototype`** in the LDAP server must exactly match the name of a user prototype in the station.
- If the **`Attr Prototype`** does not match a user prototype name or this property is blank, the system uses the **Default Prototype** as its source.

An LDAP user may be a member of multiple groups.

Kerberos and the single-sign-on feature

Niagara supports Kerberos authentication when logging in to a station. Kerberos is a widely used authentication protocol that helps to keep your credentials and station safe.

SSO (Single Sign On) is an access control feature of Kerberos that allows the automatic logging in to multiple related, but independent software systems. When you use a browser to log in with LDAP and Kerberos, you provide a single set of credentials and receive a ticket, which allows you to automatically gain access to all networked stations. SSO also makes it possible to log in to individual stations without being prompted for user name or password each time.

Logging in with Kerberos credentials

Kerberos is an open-source computer network authentication protocol that uses tickets to verify the identity of users before allowing them to access network resources.

Prerequisites:
Your client host (your PC) is part of the same LDAP realm as the station.

Step 1. Launch Workbench and open a platform or station.
If Workbench is able to acquire your native Kerberos credentials, it displays the **Authentication** window with credentials filled in.

Step 2. Do one of the following:

- If Workbench fills in your credentials with credentials from the LDAP server, click **OK** to log in.
- If Workbench is unable to acquire your native Kerberos credentials from the LDAP server, it displays a simple login window.

- To log in as a different LDAP user or as a local user, enter different credentials.

Result
You are logged in using Kerberos authentication.

Using a browser and Kerberos to log in with a single sign on

A single sign-on saves time because the system requires a user to enter credentials only once.

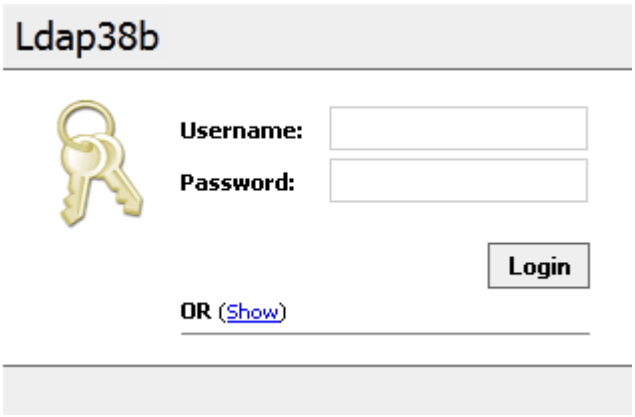
Step 1. Do one of the following:

- If the station shares the current realm, log in as the current user by clicking the realm login button.

  For this choice to work, the station must reside on the same realm that you are on. For example, if you are logged in to the FACTORY realm, the system cannot use your credentials to access a station set up for the HQ realm.
- For SSO access, go to the `/login-kerb` page (instead of this default `/login` page) and the system directly logs you in to the station without having to click a **Login** button.

Step 2. If you can successfully log in using SSO and want to bypass the login window in the future, click to select the `Remember my choice` check box at the bottom of the window.
This is effectively the same as going to the station's `/login-kerb` page.

Step 3. If you are not using the SSO feature, click the `OR (Hide)` link.
This reduces the size of the login Window to include just credentials.



Step 4. To restore the login window, click the `OR (Show)` link.
Browser cache maintains the configuration of the last-used login window.

Result

NOTE: All N4 JACE controllers support the Kerberos SSO (Single Sign On) feature from a browser, while AX controllers do not, with the exception of JACE-8000s and JACE-9000s. The reason for this is that Kerberos SSO via the browser is not supported in Java 5 (used in AX PPC JACEs), but all N4 JACEs use Java 8 (which does support it) including JACE-8000s running AX-3.8U1.

Using a browser and only LDAP credentials to log in

There may be several reasons to log in with your LDAP credentials each time you access a controller or station. One reason is that you wish to log in as a different LDAP user, or you wish to log in as a local station user (such as admin).

Step 1. Open your browser and enter the IP address for your controller in the locator field.

The system displays a login window.

Step 2. Enter your credentials and click the **Login** button.

Result

NOTE: An https secure connection is required whenever logging in with username and password for Kerberos or LDAP, and foxs connection is required when logging in with LDAP. If using a nonsecure connection you will see a login failure and a “secure connection required” message. .

Prevent Duplicate Usernames

The procedure explains how to create username using the **`Prevent Duplicate Username`** property. When the property is set true, which will cause all LDAP usernames to be lowercased during LDAP login, resulting in a case-insensitive username.

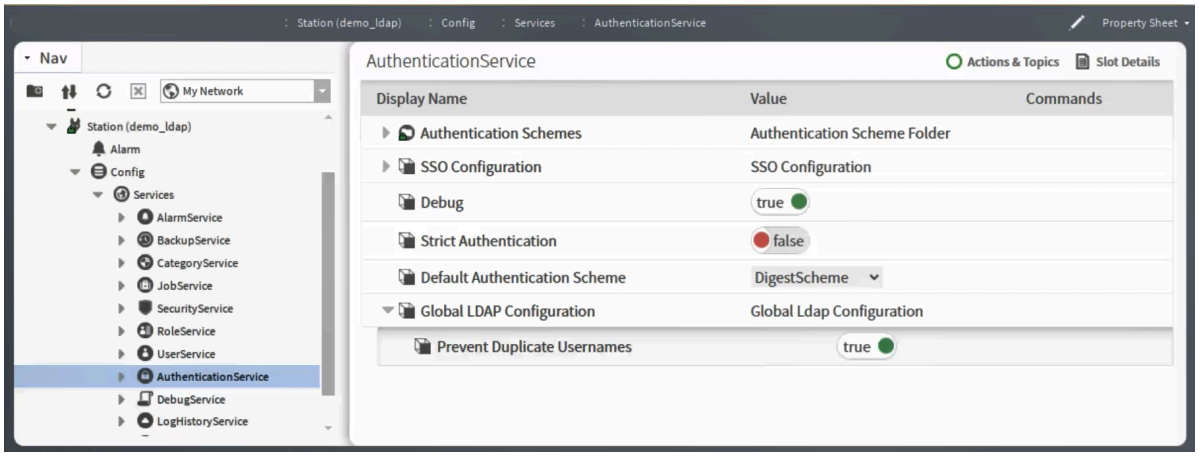
NOTE: Certain locales may run into issues lowercasing the username. It is recommended to use this feature when usernames use only ASCII characters.

You are connected to a Workbench and the station is running.

Step 1. Open the ldap palette.

Step 2. Drag the **`Ldap Scheme`** in to the station's **Config** > **Services** > **Authentication Service** > **Authentication Schemes**

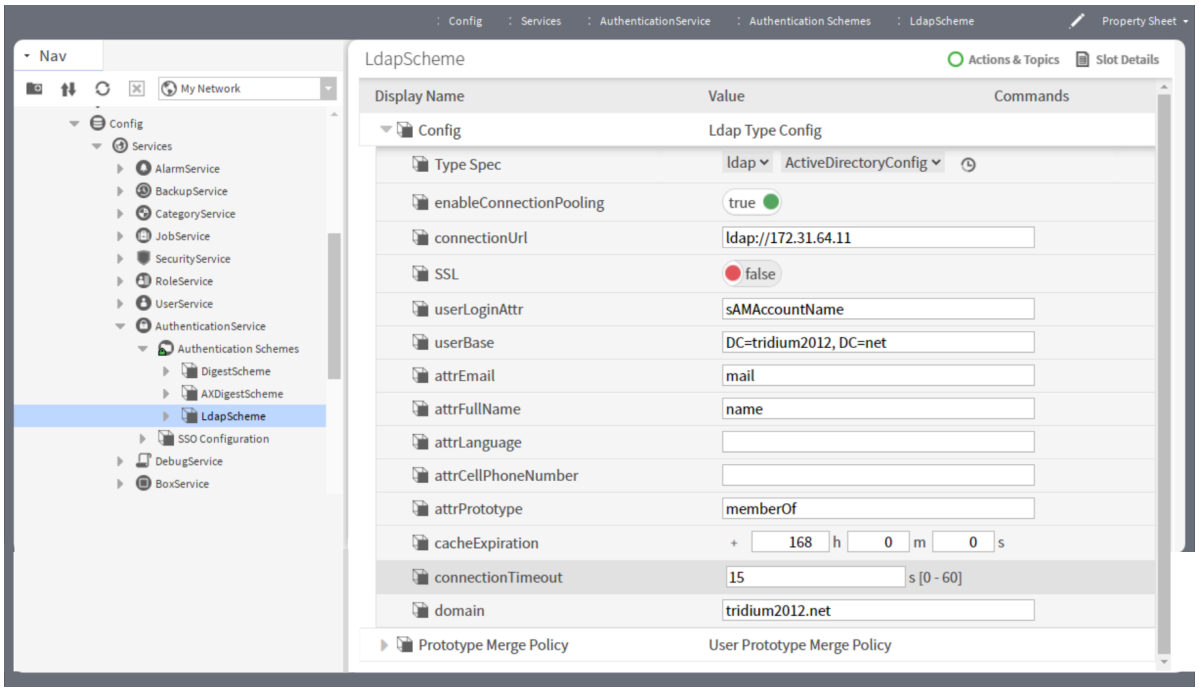
Step 3. To open the property sheet, right-click the **Authentication Service** > **Views** > **Property Sheet**. The **Property Sheet** view opens. You can see a new property **`Prevent Duplicate Usernames`**.



Step 4. Expand **Global LDAP Configuration**, set the **`Prevent Duplicate Username`** to **`true`** and click **Save**.

Step 5. To configure the LDAP Scheme to a server, right-click **LdapSchemeViewsProperty Sheet**.

The **Property Sheet** view opens.



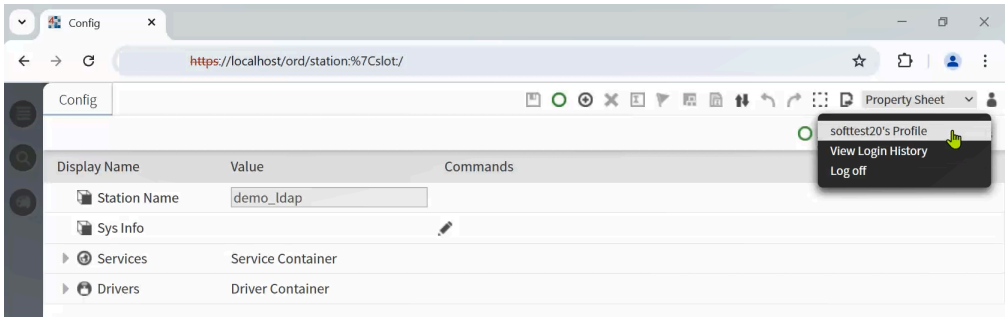
Step 6. In the **Config** property, enter the following details and click **Save**.

- For **Connection Url**, enter the URL for the LDAP.
- For **userLoginAttr**, enter user login name.
- For **userBase**, enter domain components of the domain server.
- For **attrEmail**enter the email address, for example: Jdoe@email.com
- For **attrFullName** enter the full name.
- For **attrPrototype** enter User Prototype.
- For **Domain** enter the domain name.

Step 7. Go to browser view, enter the station address and press Enter.
The browser opens a window, prompting you to authenticate yourself to the station.

Step 8. Enter the **Username**, followed by **Password** and click **Login**.

Upon clicking the button, you immediately authenticate to the station and user is created in the **UserService**.



.

Chapter 3. Components

Components include services, folders and other model building blocks associated with a module. You may drag them to a property or wire sheet from a palette.
Descriptions included in the following topics appear as context-sensitive help topics when accessed by:

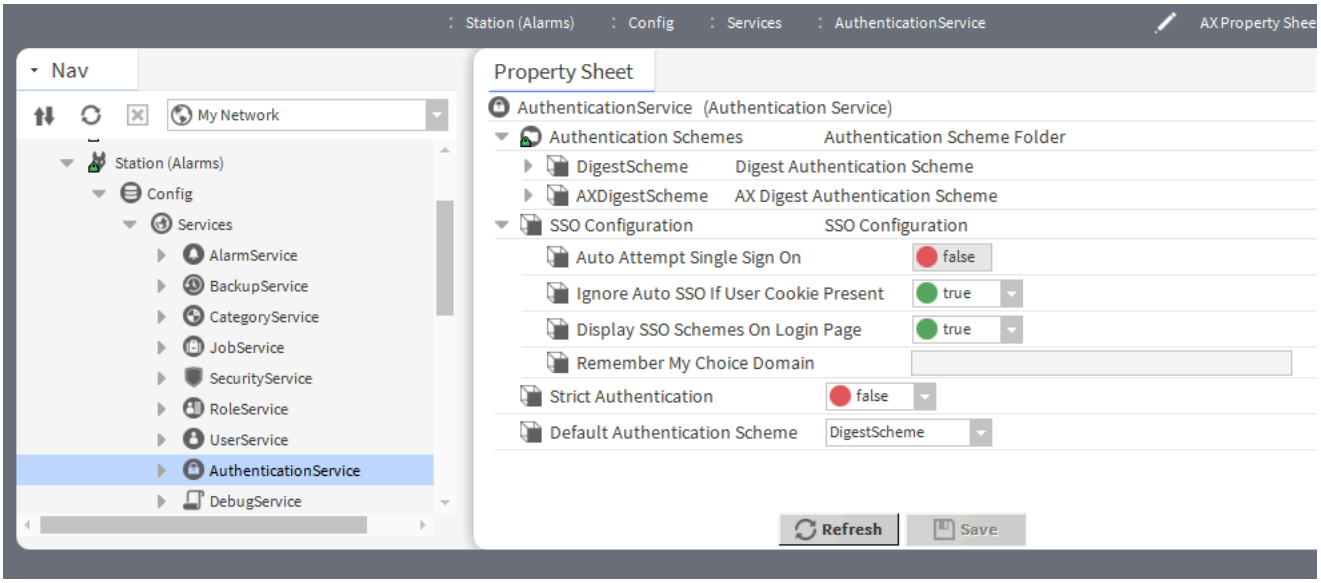
- Right-clicking on the object and selecting **Views** > **Guide Help**
- Clicking **Help** > **Guide On Target**

Authentication Service (baja-AuthenticationService)

This component manages how users verify their identity to the station, using authentication schemes. Some schemes require password configuration, others do not. The **AuthenticationService** node is located in the **Services** container.

This component is located in the baja palette.

Figure 2. AuthenticationService properties



To access, expand **Config** > **Services** and double-click **AuthenticationService**.

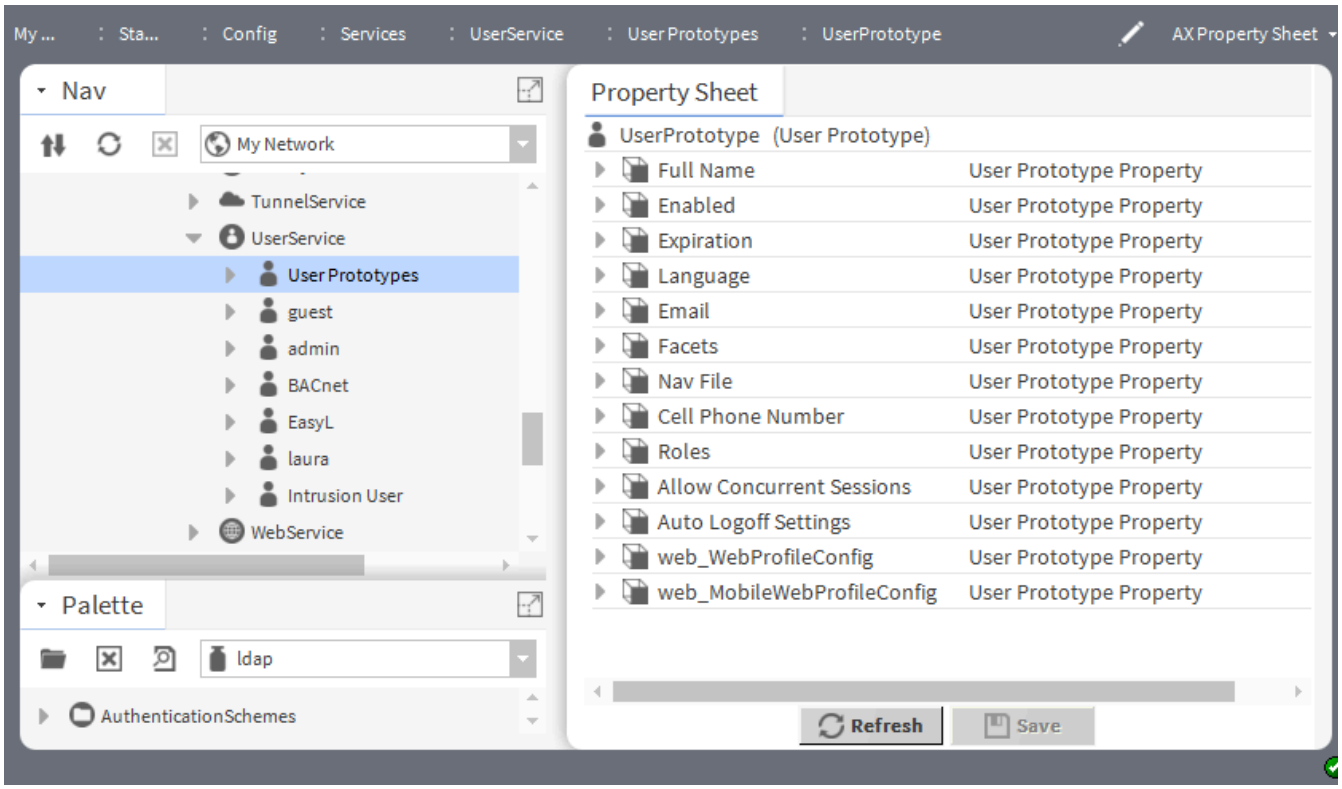
Property	Value	Description
Authentication Schemes	additional properties	Identifies the authentication scheme provided by the baja module. The **New Station** wizard installs two default authentication schemes. DigestScheme provides SCRAM-SHA256 (Salted Challenge Response Authentication Mechanism) technology for

Property	Value	Description
		connecting framework entities. Several messages are passed back and forth to prove the client knows the password. For property descriptions, refer to "baja-DigestScheme." AXDigestScheme provides compatibility with stations running a previous software version. For property descriptions, refer to "baja-DigestScheme." Additional schemes may reside in other palettes. Developers may also create authentication schemes for special circumstances. You pick the one or two schemes you wish to use, drag them from the palette and drop them directly under the AuthenticationService in the Nav tree. A topic for each individual scheme documents the properties for each.
SSO Configuration	additional properties	Enables aspects of SSO (Single Sign-On) functionality, such as whether or not to automatically attempt single sign on when users log on to a station. "SSO Configuration (baja-SSOConfiguration)" documents the additional properties.
Strict Authentication	`true` or `false` (default)	Enforces authentication controls.
Default Authentication Scheme	drop-down list	Selects the default authentication scheme to use.

baja-UserPrototype

This component controls how LDAP users are created and where their properties come from.

Figure 3. UserPrototype properties



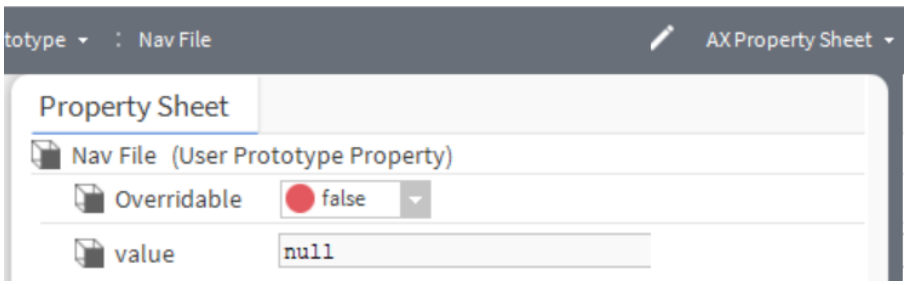
To view these properties, expand **Config** > **Serivces** > **UserService** > **UserPrototype** and double-click one of the user properties (**UserPrototype** may have a different name; user properties are, for example, **`Full Name`**, **`Language`**, **`Roles`**, etc.).

Each of these properties has two sub-properties. The **baja-UserPrototypeProperty** describes these properties.

baja-UserPrototypeProperty

This component represents a user attribute (property), such as full name, email address, cellphone number, etc.

Figure 4. UserPrototypeProperty properties



To access these properties, expand **Config** > **Drivers** > **UserPrototype** and double-click a user property. Each

user property has same properties and descriptions.

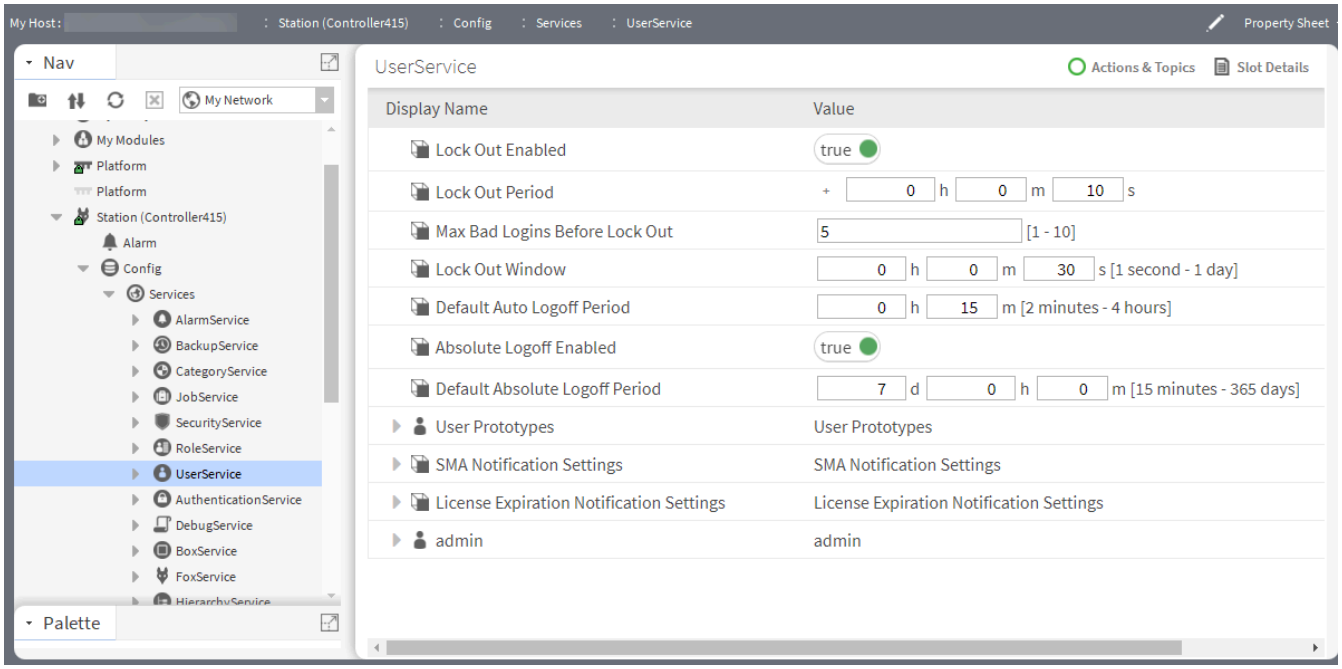
Name	Value	Description
Overridable	`true` or `false` (default)	Determines whether or not the property can be manually overridden on an LDAP user that was created from this prototype. `true` configures the property to be manually overwritten. `false` configures the property as read-only. Although a user may change it, the next time the user logs in to the server the value of the property returns to its original value.
Value	string (defaults to null)	Determines what value to set for the matching property on the user when creating or updating from a prototype.

UserService (baja-UserService)

This service manages all system users: human and machine. You access it by right-clicking **UserService** and clicking **Views** > **Property Sheet**.

The **User Manager** is the primary view of this service. By default, creating a new station using the **New Station** wizard includes the **UserService**. The baja module makes this service available.

Figure 5. User Service property sheet view



To access these properties, expand **Config** > **Services**, right-click **UserService** and click **Views** > **AX Property Sheet**.

Property	Value	Description
Lock Out Enabled	`true` or `false`	
Lock Out Period	`true` or `false`	
Max Bad Logins Before Lock Out	Number from 1–10 (defaults to 5)	
Lock Out Window	hours minutes seconds (defaults to 30 seconds)	
Default Auto Logoff Period	0000h 15m (default)	Specifies the amount of time that a period of inactivity may last before a station connection is automatically disconnected. The acceptable range of values is two minutes to four hours. This limit is observed only when the User's **`Use Default Auto Logoff Period`** property is set to `true`.
Absolute Logoff Enabled	`true` (default) or `false`	If enabled, the **`Absolute Logoff Enabled`** property is added to the **User Manager** view where it can be applied to individual users (as of Niagara 4.15).

Property	Value	Description
Default Absolute Logoff Period	Number from 15min to 365 days (defaults to 7 days)	Specifies the amount of absolute time regardless of user activity or inactivity before a station connection is automatically disconnected. This limit is observed only when the user's Absolute Logoff Enabled property is set to `true`.
SMA Notification Settings	multiple properties	Configures the SMA (Software Maintenance Agreement) whose properties are documented in baja-SMANotificationSettings, which is in the document.
User Prototypes	multiple properties	Serves as a container for the default and other user prototypes whose properties are documented in baja-UserPrototypes, which is in the document.

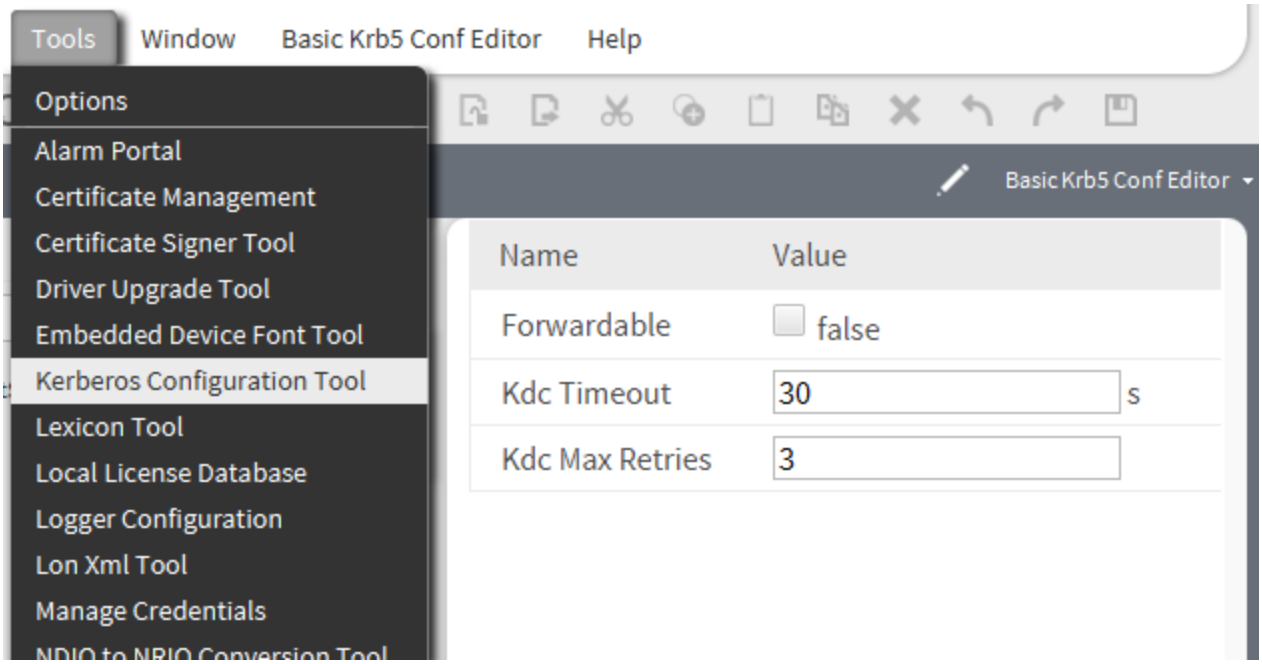
ldap-KerberosConfigurationTool

In Niagara, two editors provide simple text editors, which you can use to manually edit an existing Kerberos configuration file (`krb5.conf`) or to create a new one.

Basic Krb5 Conf Editor

Kerberos authentication requires the ability to acquire Kerberos tickets that can be forwarded. The editor allows you to enable and disable the **`Forwardable`** property.

Figure 6. Basic Krb5 Conf Editor



Property	Value	Description
Forwardable	`true` (default), `false`	Enables and disables forwarding of Kerberos tickets.
Kdc Timeouts	30 (default)	Required for redundant server support, specifies the length of time the station attempts to connect to the key distribution center before failing the connection attempt.
Kdc Max Retries	3 (default)	Required for redundant server support, specifies the maximum number of times the station attempts to connect to one key distribution center before to the next one.

NOTE: Values entered for the **`Kdc Timeouts`** and **`Kdc Max Retries`** properties should be tailored to your specific scenario based on how long successful kdc connections generally take and when to configure the the cut-off time after which the connection is considered to have failed. As with the connection timeout above, this time needs to be not too short to cause false connection failures, but not so long as to cause excessive delays when a server is down.

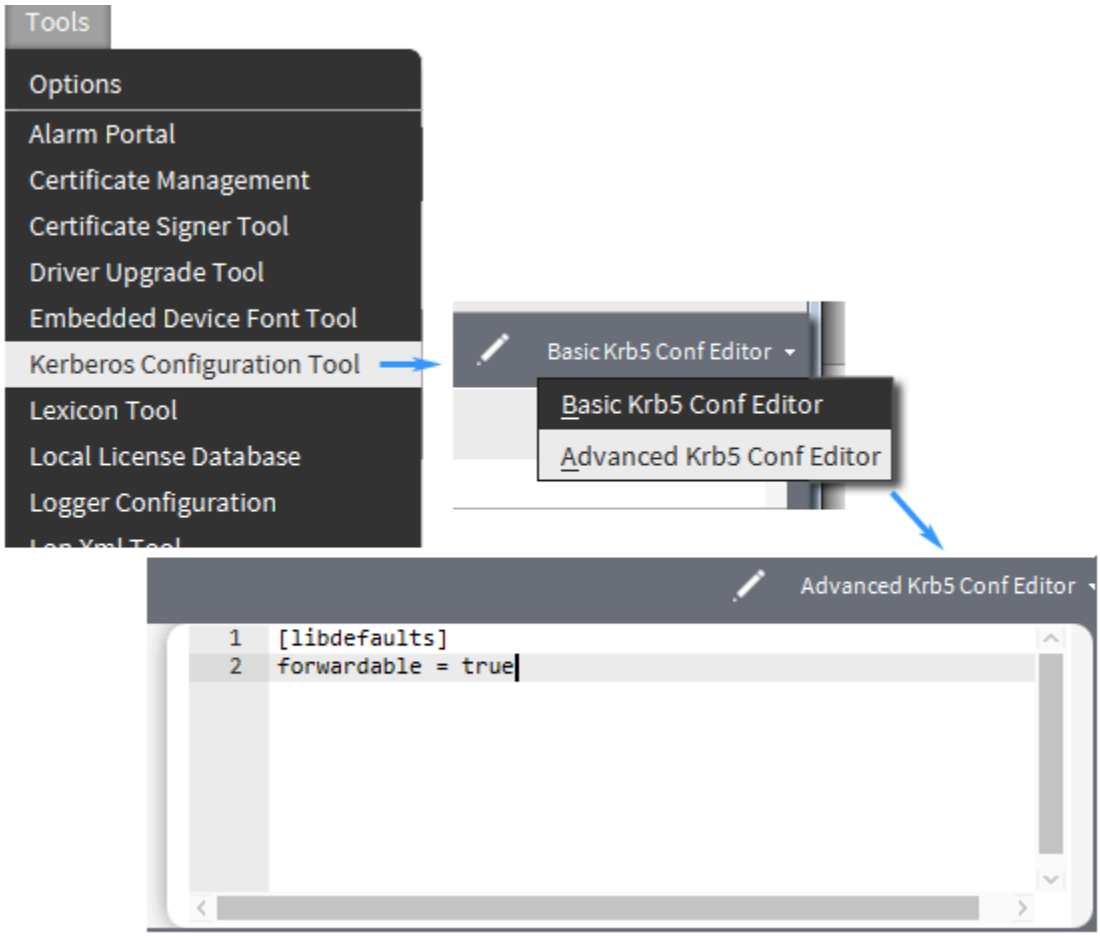
Advanced Krb5 Conf Editor

On a Windows host, the primary location for the file is: `NIAGARA_HOME/security/krb5.conf`. Only if this file is missing would you fall back to the Java krb.conf or operating system specific `krb.conf/ini`.

On a Linux host the file location is: `/etc/krb5.conf`.

NOTE: If you are working with Linux, some systems may require a more advanced `krb5.conf` file. If that is the case, have your Kerberos administrator set up this file for you.

Figure 7. Advanced Krb5 Conf Editor



The file requires only the two lines contained in this view.

ldap-LdapAuthenticationScheme

Adding the LdapScheme manages Niagara 4 user authentication using an LDAP (Lightweight Directory Access Protocol) server. This allows you to connect to a previously existing database of users—a huge advantage when setting up new users (you don't have to manually create new users in each station). The LDAP server also keeps passwords centralized and in sync.

Common properties

One common example of an LDAP server is ActiveDirectory, which is used by Windows to manage users.

NOTE: TLS is required for LDAP authentication. If an LDAP user attempts to login over a nonsecure connection, a login failure occurs with a message stating "Secure connection required". Enable TLS secure communication in the FoxService (**`Foxs`** `enabled`) and WebService (**`Https`** `enabled`). Additionally, if the LdapScheme is not set to `Ldap V3` with either the CRAM-MD5 or DIGEST-MD5 authentication mechanism, the system sends the username and password to the LDAP server in plain text. Again, ensure that TLS is enabled in the LdapScheme. This may require you to configure the LDAP server to support communication security (SSL/TLS).

Property	Value	Description
Type	drop-down list of configuration types	Selects the type of configuration. The system supports sets of configuration properties: • `Active Directory Config` • `Ldap V2 Config` • `Ldap V3 Config` Each type supports slightly different properties. Choose the type that best fits your Ldap server's requirements.
Enable Connection Pooling	`true` (default) or `false`	Enables (`true`) and disables (`false`) the use of a connection pool. To speed processing, LDAP servers maintain a pool of connections. A request from the system that uses an existing connection saves valuable processing time, which improves system performance. Do not change the default (true = enabled) setting unless you know what you are doing.
Connection URL	ldap://<your.domain.net> orldap://<your.domain.net:nnn>	Identifies the URL (<your.domain.net>) for the LDAP server. Standard LDAP ports are 389, or 636 (if using SSL). If the server uses a non-standard port, include the port (<your.domain.net:nnn>) in the URL, for example, <ldap://your.domain.net.999>.
SSL	`true` or `false`	Enables (`true`) and disables (`false`) secure communication. If set to `true`, make sure that SSL (3.8) or TLS (4.0) is enabled in the station's FoxService (for Workbench-to-station access) and WebService (for browser-to-station access). Note that in FoxService and WebService TLS must be enabled whether SSL is true here or not.
User Login Attr	textFor AD this value defaults to `sAMAccountName`	Identifies the specific attribute in the LDAP directory to store the LDAP user login name. For AD servers, this is always `sAMAccountName`. For OpenLDAP servers, it would be `uid`.
User Base	domain components	Identifies the sub-tree of the LDAP server in which users who can access this station are found. At the very

Property	Value	Description
		least it must contain the domain components of the server's domain, for example: `DC=domain , CD=net .`
Attr Email	Email address (AD defaults to: `mail`)	Identifies the specific attribute in the LDAP directory to store the user's LDAP email address. This value populates the Niagara user's **`Email`** property.
Attr Full Name	text (The AD defaults to: `name`)	Identifies the specific attribute in the LDAP directory to store the user's full name. This value populates the Niagara user's **`Full Name`** property.
Attr Language	two-letter language code (AD defaults to blank)	Identifies the specific attribute in the LDAP directory to store the user's language. This value populates the Niagara user's **`Language`** property.
Cell Phone Number	telephone number (AD defaults to mobile)	Identifies the attribute in the LDAP directory that stores the user's mobile phone number. This value populates the Niagara user's **`Cell Phone Number`** property.
Attr Prototype	text (AD defaults to `memberOf`).	Identifies the **`User Prototype`** with which the system populates a new user's local properties. If this property is blank or the name does not match any user prototype, the system uses the **`Default Prototype`** to populate local user properties. If a user belongs to multiple user groups (user prototypes), the top-to-bottom order of prototypes determines which prototype the system uses. If the value of a user prototype property changes, the system dynamically updates user properties accordingly.
Cache Expiration	date and time	Defines a future date after which the system no longer stores a user's password in cache. When an LDAP server is unavailable a user can still log on with the cached credentials until this date and time.

Property	Value	Description
		This property applies to Kerberos authentication even though the station never receives the user's password. Instead, the station verifies the corresponding Kerberos user ticket against the cached user information.
Connection Timeout	time	Determines the length of time the station attempts to connect to the LDAP server before the connection fails. The station will not fail over to the next LDAP server until the first connection attempt is unresponsive for the amount of time specified in the connection timeout. This time should not be too short to cause false connection failures, but not so long as to cause excessive delays when a server is down.

Active Directory Config

This property is unique to Active Directory.

Property	Value	Description
Domain	text	Supplies the domain name used to contact the LDAP server.

LDAP V2 Config

These properties are unique to LDAP V2 Config.

Property	Value	Description
Domain	text	Supplies the domain name used to contact the LDAP server.
Connection Pwd	password	Defines the password for the user specified in property `Connection User`. When used, requires a valid password in the LDAP server. The system uses this password to connect to the server for authentication.

LDAP V3 Config

These properties are unique to LDAP V3 Config.

Property	Value	Description
Bind Format	BFormat (Baja Format) syntax with a default value of `<%userName%>`	Specifies how to send the user name to the server. This feature applies to Ldap V3 only. Every LDAP server is different. For the most part, a user base and logon name are sufficient to find a user in the LDAP directory. However, when using `DIGEST` authentication, it may be necessary to specify the exact format of the logon name to send to the server. In Active Directory (AD) 2000, this might be: `%username%@domain.com`. Later versions of AD would reject this format, however, they would accept a username based on how the server stores passwords. **`Bind Format`** allows you to specify how to send the name to the server. For example, using a BFormat, this property would be: `%username%@domain.net` or `cn=%username,%userBase%`. For details, see the engineering notes document, *BFormat (Baja Format) Property Usage*. **NOTE:** If the value of this property needs to be changed, consult with your onsite LDAP administrator for assistance .
Connection User	text	Defines 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). When used, requires a valid user name in the LDAP server. The system uses this name to connect to the server for authentication.
Connection Pwd	password	Defines the password for the user specified in property **`Connection User`**. When used, requires a valid password in the

Property	Value	Description
		LDAP server. The system uses this password to connect to the server for authentication.
Authentication Mechanism	dropdown list	LDAP v3 supports several methods for user validation. These are known as SASL (Simple Authentication and Security Layer) mechanisms. `None` `Simple` (default) sends the user name and password to the server in clear text. `CRAM-MD5` obscures the password for security. `DIGEST-MD5` obscures the password for security.

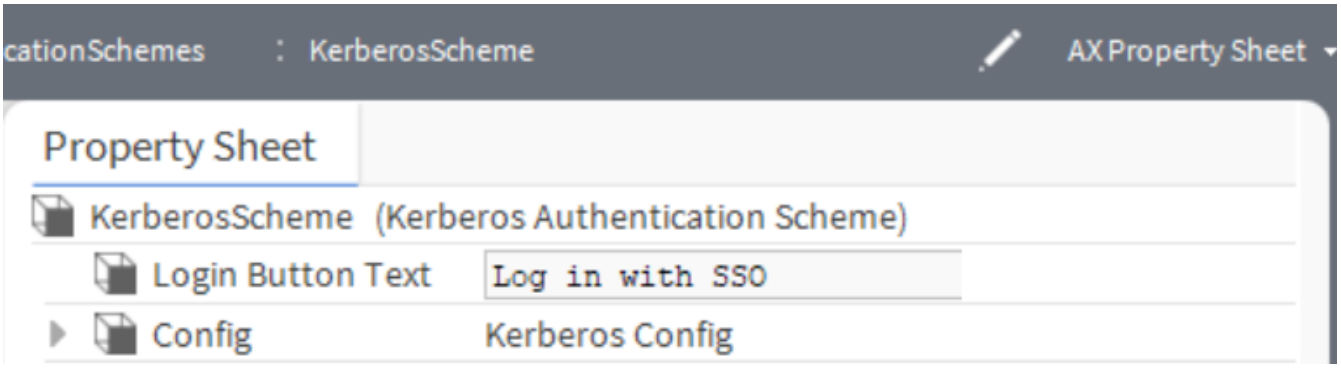
ldap-KerberosAuthenticationScheme

This component provides Kerberos authentication for logging in users to a station.

Kerberos is a network authentication protocol that allows nodes communicating over a network that is not secure to prove their identity to one another in a secure manner. Aimed primarily at a client–server model, it provides mutual authentication—both the user and the server verify each other's identity.

To access this property, In a Nav tree, expand **Config** > **Services** and double-click **KerberosScheme**.

Figure 8. Kerberos authentication properties

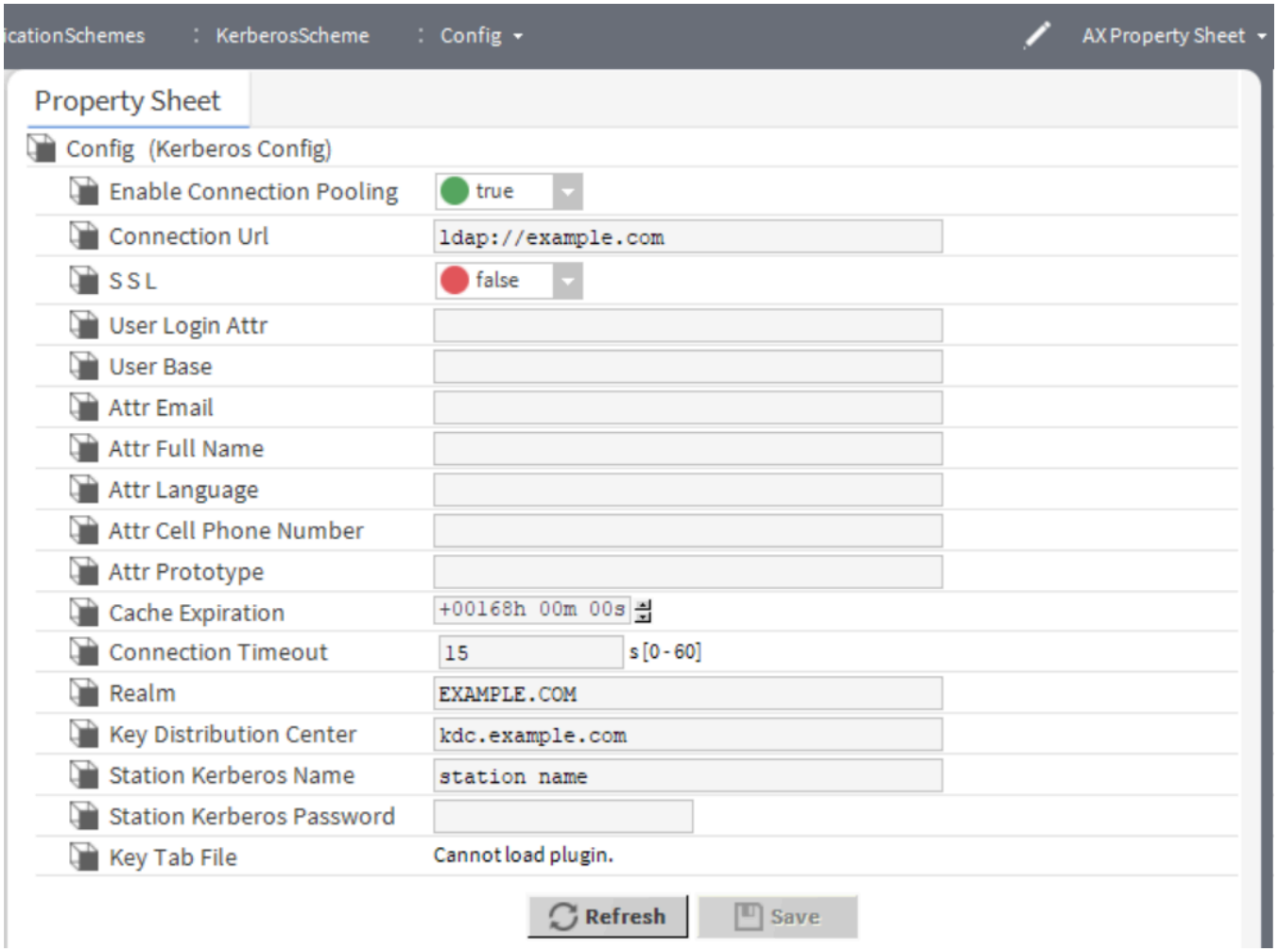


Property	Value	Description
Login Button Text	text	

ldap-KerberosConfig

LDAP and Kerberos together make for a great combination. Kerberos manages credentials securely (authentication) while LDAP stores authoritative information about the accounts, such as what they are allowed to access (authorization), the user's full name and uid. You can add helpful things, such as an external email address or a room number in a structured way.

Figure 9. Kerberos Config properties



To access these properties, expand **Config** > **Services** > **KerberosScheme** and double-click **Config**.

Property	Value	Description
Enable Connection Pooling	`true` (default) or `false`	Enables (`true`) and disables (`false`) the use of a connection pool. To speed processing, LDAP servers maintain a pool of connections. A request from the system that uses an existing connection saves valuable processing time, which improves system performance. Do not change the

Property	Value	Description
		default (true = enabled) setting unless you know what you are doing.
Connection URL	ldap://<your.domain.net> orldap://<your.domain.net:nnn>	Identifies the URL (<your.domain.net>) for the LDAP server. Standard LDAP ports are 389, or 636 (if using SSL). If the server uses a non-standard port, include the port (<your.domain.net:nnn>) in the URL, for example, <ldap://your.domain.net.999>.
SSL	true or false	Enables (true) and disables (false) secure communication. If set to true, make sure that SSL (3.8) or TLS (4.0) is enabled in the station's FoxService (for Workbench-to-station access) and WebService (for browser-to-station access). Note that in FoxService and WebService TLS must be enabled whether SSL is true here or not.
User Login Attr	textFor AD this value defaults to sAMAccountName	Identifies the specific attribute in the LDAP directory to store the LDAP user login name. For AD servers, this is always sAMAccountName. For OpenLDAP servers, it would be uid.
User Base	domain components	Identifies the sub-tree of the LDAP server in which users who can access this station are found. At the very least it must contain the domain components of the server's domain, for example: DC=domain, CD=net.
Attr Email	Email address (AD defaults to: mail)	Identifies the specific attribute in the LDAP directory to store the user's LDAP email address. This value populates the Niagara user's **Email** property.
Attr Full Name	text (The AD defaults to: name)	Identifies the specific attribute in the LDAP directory to store the user's full name. This value populates the Niagara user's **Full Name** property.
Attr Language	two-letter language code (AD defaults to blank)	Identifies the specific attribute in the LDAP directory to store the user's language. This value populates the Niagara user's **Language** property.

Property	Value	Description
Attr Cell Phone Number	telephone number (AD defaults to mobile)	Identifies the attribute in the LDAP directory that stores the user's mobile phone number. This value populates the Niagara user's **`Cell Phone Number`** property.
Attr Prototype	text (AD defaults to `memberOf`).	Identifies the **`User Prototype`** with which the system populates a new user's local properties. If this property is blank or the name does not match any user prototype, the system uses the **`Default Prototype`** to populate local user properties. If a user belongs to multiple user groups (user prototypes), the top-to-bottom order of prototypes determines which prototype the system uses. If the value of a user prototype property changes, the system dynamically updates user properties accordingly.
Cache Expiration	date and time	Defines a future date after which the system no longer stores a user's password in cache. When an LDAP server is unavailable a user can still log on with the cached credentials until this date and time. This property applies to Kerberos authentication even though the station never receives the user's password. Instead, the station verifies the corresponding Kerberos user ticket against the cached user information.
Connection Timeout	time	Determines the length of time the station attempts to connect to the LDAP server before the connection fails. The station will not fail over to the next LDAP server until the first connection attempt is unresponsive for the amount of time specified in the connection timeout. This time should not be too short to cause false connection failures, but not so

Property	Value	Description
		long as to cause excessive delays when a server is down.
Realm	UPPERCASE lettersEXAMPLE.COM	Identifies the system on which the LDAP server resides. You get this information from your Kerberos administrator.
Key Distribution Center	text, for example: kd.example.com	Specifies the name of the Kerberos Key Distribution Center that the system contacts to get a ticket, which, like a key, is used to authenticate the user to the Niagara system. You get this information from your Kerberos administrator.
Station Kerberos Name	text	As part of securely delegating Kerberos tickets, this property represents the station as a user in the Kerberos database. If logging in only via Workbench, this user can be any user or service in the Kerberos directory. However, if the user logs in via a browser, the user must be a service in the form: HTTP/service-Name.domain.com, where serviceName.domain.com is how the station is to be accessed in the browser, (for example, http://stationkerb1.mydomain.com). The service name for the station Kerberos name typically omits a bit of the normal http URL syntax, for example: http/ stationkerb1.mydomain.net instead of http://stationkerb1.mydomain.net. You may need to ask the Kerberos administrator to create the service for you in the Kerberos database. **NOTE:** 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.

Property	Value	Description
Station Kerberos Password	text (defaults to blank)	Specifies the password for the Kerberos station user identified by the **Station Kerberos Name** property. If you are using a keytab file, you can leave this property blank.
Key Tab File	file name	Defines the keytab file that contains a key table. Kerberos services usually do not use a password to authenticate. Instead, they use a file. To authenticate 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. You must copy that keytab file to this secure location on the Niagara 4 platform: `protected_station_home/ldap`. You need to create the `ldap` directory manually. For the **KeyTab File** property, select the keytab file from the drop-down. Again, if you are using a keytab, you can leave the **Station Kerberos Password** property blank (default).

Chapter 4. Glossary

The following glossary entries relate specifically to the topics that are included as part of this document.

To find more glossary terms and definitions refer to glossaries in other individual documents.

Alphabetical listing

LDAP user
A user whose access permissions are managed by an LDAP (Lightweight Directory Access Protocol) server.

local user
A user that has been set up using the standard Niagara user properties. The system cannot validate a local user against the LDAP database. This type of user can only log in to a local host. Two local users are common: the admin user and a service user. The admin user logs in to the host during initial setup and rarely thereafter to update EC-Net software. A service user represents the host to other remote users.

realm
A set of Niagara Station devices that share the same Kerberos database. The Kerberos database resides on the LDAP server.

service user
A user created within a station for the sole purpose of representing the station as the client of another remote host. No one ever logs in to host as the service user. It represents the host to other remote hosts on the **NiagaraNetwork** when configuring the remote host's **`Client Connection`** properties under the device.

super user
A type of local user that has full read-write permissions within a station. This user can set up and update station software as well as create users and manager access permissions. The default admin user is an example of a super user.