

# KNIME Server Erweitertes Setup

## Leitfaden

KNIME AG, Zürich, Schweiz

Version 4.18 (letzte Aktualisierung auf )



## Inhaltsverzeichnis

<a href="#page2" style="color: #000000; text-decoration: underline;">Einleitung . . . . .<a href="#"<a href="#page3" style="color: #000000; text-decoration: underline;">Enterprise User Authentication . . . . .<a href="#"<a href="#page3" style="color: #000000; text-decoration: underline;">Konfigurieren einer LDAP Verbindung . . . . .<a href="#"<a href="#page17" style="color: #000000; text-decoration: underline;">Konfigurieren von Single Sign On . . . . .<a href="#"<a href="#page32" style="color: #000000; text-decoration: underline;">Dynamische Profile für Benutzer . . . . .<a href="#"<a href="#page33" style="color: #000000; text-decoration: underline;">OpenID Connect Authentication . . . . .<a href="#"<a href="#page33" style="color: #000000; text-decoration: underline;">Authentication Value Konfiguration . . . . .<a href="#"<a href="#page39" style="color: #000000; text-decoration: underline;">KNIME Server-Client Konfiguration . . . . .<a href="#"<a href="#page42" style="color: #000000; text-decoration: underline;">Debugging OIDC Authentication . . . . .<a href="#"<a href="#page43" style="color: #000000; text-decoration: underline;">Mit Ihrer eigenen Tomcat installieren . . . . .

# Einleitung

Dieser Leitfaden umfasst erweiterte Themen einer KNIME Server-Bereitstellung, Einrichtung und Konfiguration in eine Unternehmensumgebung.

Wenn Sie den KNIME Server installieren möchten, sollten Sie zuerst die [KNIME Server Installationsanleitung](#).

Für Anleitungen zum Anschluss an KNIME Server von der KNIME Analytics Platform oder mit KNIME WebPortal verweist auf folgende Anleitungen:

- [KNIME Benutzerhandbuch des Servers](#)
- [KNIME WebPortal Benutzerhandbuch](#)

Für alle regulären Administrationskonfigurationsoptionen und ein grundlegendes Verständnis von KNIME Server bitte konsultieren [KNIME Leitfaden für die Verwaltung von Servern](#).

Im Folgenden wird angenommen, dass Sie ein Wissen über alle Themen, die in der zuvor erwähnte Führungen.

# Enterprise User Authentication

Benutzeroauthentifizierung in einer Unternehmensumgebung wird in der Regel durch einige zentralisierte Service. Der am häufigsten verwendete Service ist LDAP. LDAP-Authentifizierung ist die empfohlene Authentifizierung in jedem Fall, in dem ein LDAP-Server verfügbar ist. Wenn Sie mit Ihrem vertraut sind LDAP-Konfiguration können Sie die Details während der Installation hinzufügen oder die Datei post Installation. Wenn Sie mit Ihren LDAP-Einstellungen nicht vertraut sind, müssen Sie möglicherweise Kontakt aufnehmen Ihr LDAP-Administrator oder die Konfigurationsdetails für jedes andere Tomcat-basierte System verwenden in Ihrer Organisation. Dieser Abschnitt beschreibt die Einrichtung von KNIME Server für LDAP Authentifizierung.

Server.xml

Eine weitere Möglichkeit der Benutzeroauthentifizierung ist Single-sign-on. KNIME Server kann konfiguriert werden zur Unterstützung der Kerberos-Authentifizierung in Kombination mit LDAP. Dieser Abschnitt enthält auch Schritte für ein einfaches Kerberos Setup.

## Konfigurieren einer LDAP-Verbindung für KNIME Server

KNIME Server verwaltet alle Benutzeroauthentifizierung durch die eingebauten Mechanismen von Apache Tomcat. Daher ist die umfassendste Dokumentation zur Konfiguration der Authentifizierung die [Apache Tomcat Realm Konfiguration HOW-TO](#) (auch Active Directory) Konfiguration, siehe Abschnitt [JNDIRealm](#).

. Speziell für Informationen über LDAP

## Terminologie

In diesem Dokument beziehen wir uns auf die Einrichtung einer LDAP-Verbindung, LDAP-Konto usw. Da eine der beliebtesten Möglichkeiten zur Verwaltung der Benutzeroauthentifizierung ist Microsoft Active Directory, die LDAP unterstützt, können Sie ersetzen wollen LDAP-Konto für Aktiv Katalog-Account.

## Schnellstart

In den meisten Fällen sollte es möglich sein, Ihr lokales LDAP/Active Directory zu kontaktieren Verwalter; sie sollten in der Lage sein, die erforderlichen Informationen bereitzustellen.

Sie können folgende Fragen stellen:

ANHANG Haben sie bereits Konfigurationsdetails für einen Tomcat Server? Wenn ja, diese Verbindung Informationen können wieder verwendet werden.

2. LDAP Verbindungsinformationen (Hostname, Port, wird TLS/SSL verwendet?).
3. Ob sie Bind-Modus verwenden, oder Vergleichsmodus.

L 347 vom 20.12.2013, S. 1). Wie die Gruppeninformationen gespeichert werden.

Sie müssen Konfiguration bereitstellen, die in eine solche Vorlage passen kann:

```
VerbindungURL = "ldap://localhost:389"  
UserPattern = "uid = {0},ou = people,dc = mycompany,dc = com"  
roleBase = "ou = groups,dc = mycompany,dc = com"  
RolleName = "cn"  
RolleSearch = "(uniqueMember = {0})"  
>
```

Diese Informationen werden dem **Server.xml** Datei, die in  
tomcat>/conf/server.xml .

Für die Änderungen an der  
Konfigurationsdatei zur Wirkung.

#### Erweiterte Fehlerbehebung

Die übrigen Abschnitte dieser Dokumentation beschreiben, wie eine LDAP-Verbindung für KNIME Server. Dies ist nur ein Weg, um verwandte Informationen an einen Ort zu sammeln, nicht als umfassende Dokumentation für LDAP oder Tomcat.

Die erste Voraussetzung ist Apache Directory Studio oder ein anderes LDAP-Konfigurationstool. Wir

Verwendun  
g [Apache Directory Studio](#) die Prüfung durchzuführen. Der Vorteil dieses Tools ist, dass es offen ist source, kostenlos herunterladen, arbeitet unter Windows/Linux/Mac, so dass ein Kunde kann herunterladen Software und fragen, um zu starten.

Wir werden drei grundlegende Schritte folgen:

ANHANG LDAP Verbindungsinformationen (Hostname, Port, SSL?).

2. Ob sie Bind-Modus verwenden, oder Vergleichsmodus.
3. Wie die Gruppeninformationen gespeichert werden.

LDAP Verbindungsinformationen (Hostname, Port, SSL)

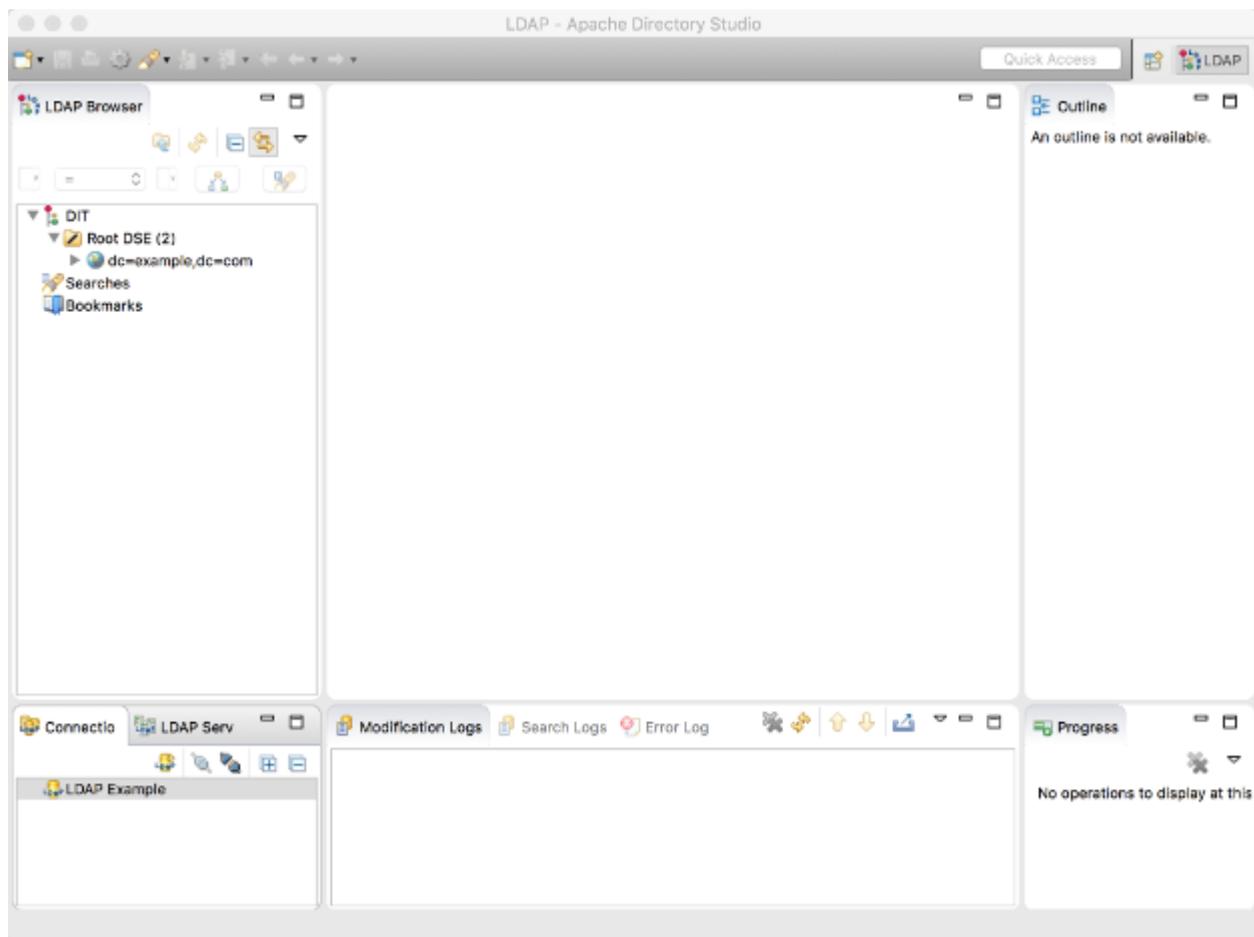
Um eine Verbindung zu einem LDAP-Server herzustellen, müssen Sie wissen:

- Der LDAP Server Hostname (oder IP)
- Ob der Server SSL gesicherte Verbindungen nutzt oder nicht

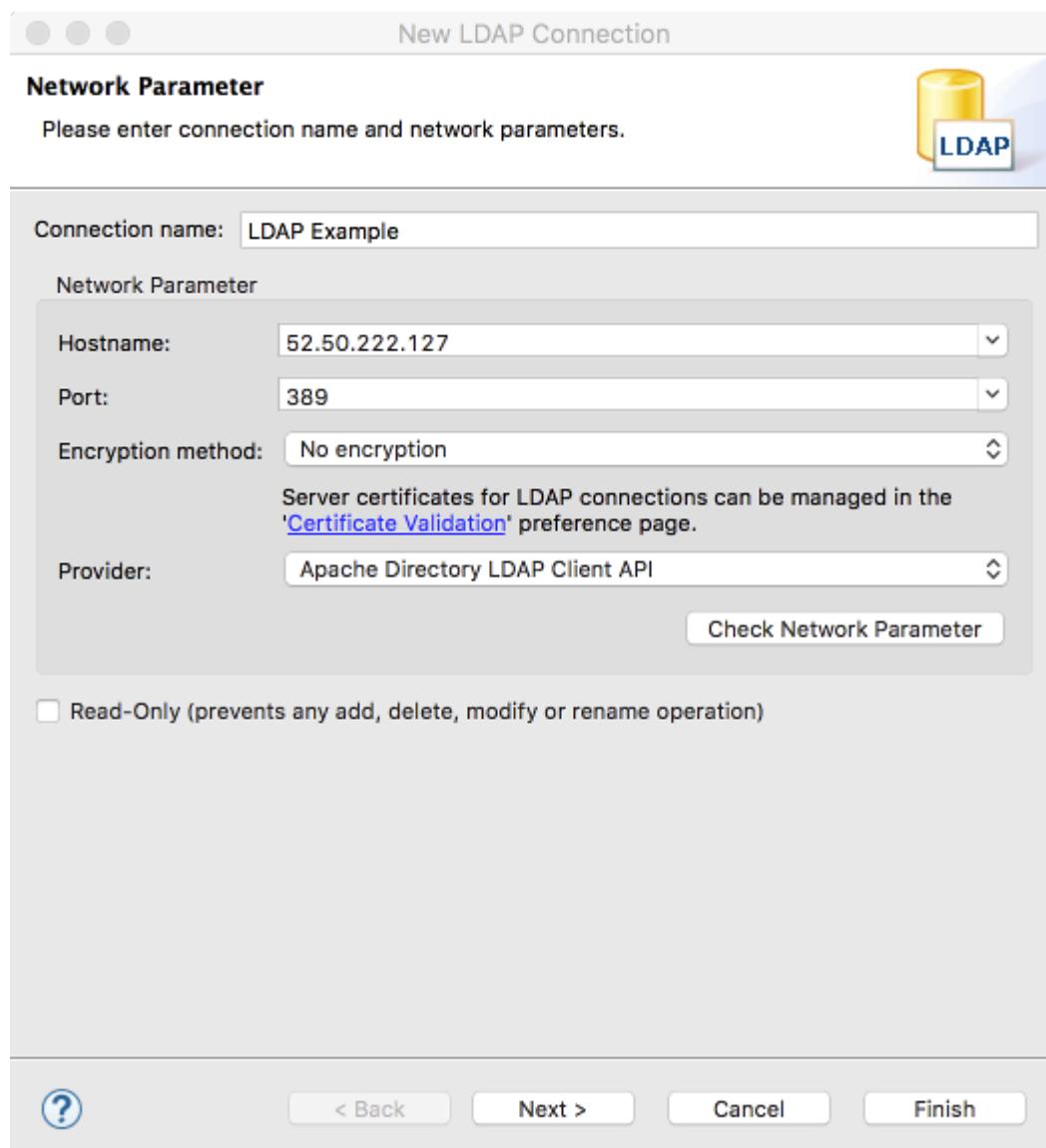
- Welcher Port verwendet wird — Standardports  
TLS) und 636 für LDAPS (SSL gesichert)
- 389 für LDAP (unverschlüsselt oder verschlüsselt)

Apache Directory Studio einrichten, um Ihr LDAP-Verzeichnis zu durchsuchen

Verbindungsaufbau zum Server

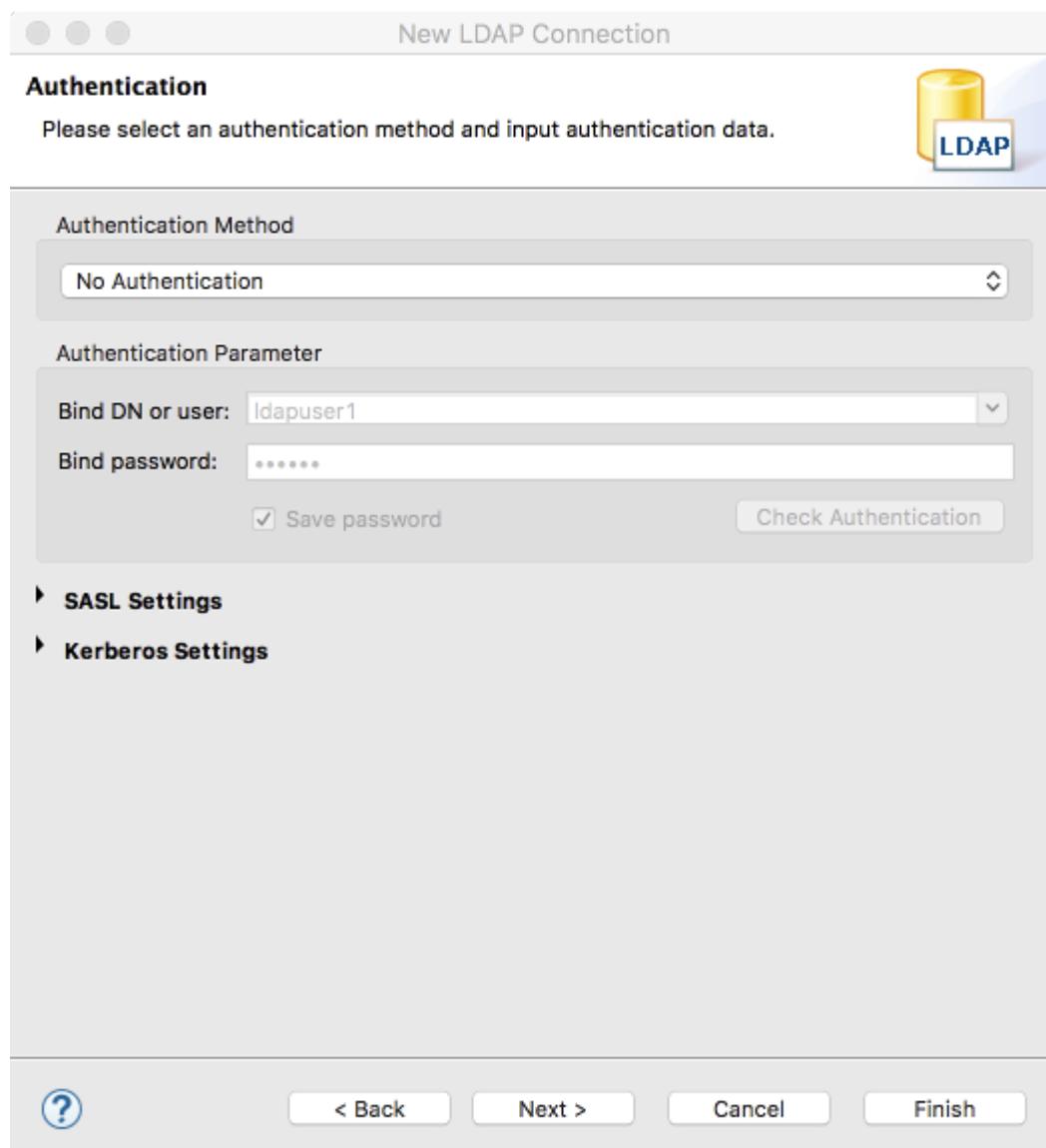


Fügen Sie in den Verbindungsdaten Ihres LDAP-Servers hinzu



Verbindungsauflauf zum LDAP-Server

Beachten Sie, dass wir hier keine Authentifizierung verwenden. In der Regel müssen Sie authentifizieren, und in die meisten Fälle kann dies Ihr LDAP Benutzername und Passwort sein.



#### Verbindungsaufbau

Sie können auf 'Fetch Base DNs' klicken, um die Antworten zu autopopulieren. In unserem Beispiel ist die Basis DN

dc=Beispiel,dc=com . Dies wird variieren, z.B. knime.com könnte die Basis DN verwenden

dc=knime,dc=com .

New LDAP Connection

### Browser Options

You can specify additional parameters for browsing the directory.



**Base DN**

Get base DNs from Root DSE Fetch Base DNs

Base DN: dc=example,dc=com

**Limits**

Count Limit:  Time Limit (s):

**Aliases Dereferencing**

Finding Base DN  Search

**Referrals Handling**

Follow Referrals manually  Follow Referrals automatically  Ignore Referrals

**Controls**

Use ManageDsAIT control while browsing  
 Fetch subentries while browsing (requires additional search request)  
 Paged Search Page Size:   Scroll Mode

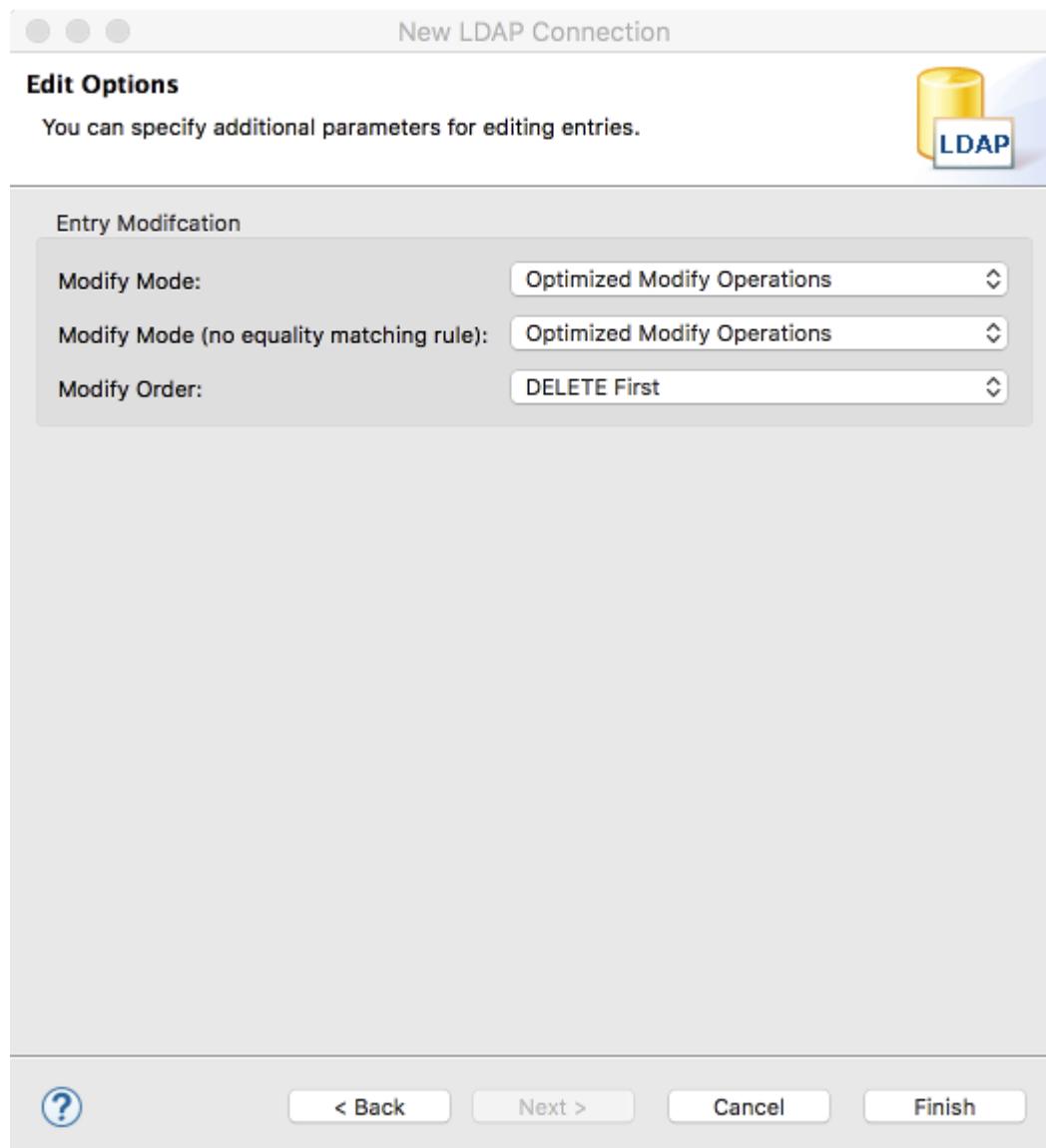
**Features**

Fetch operational attributes while browsing

? < Back Next > Cancel Finish

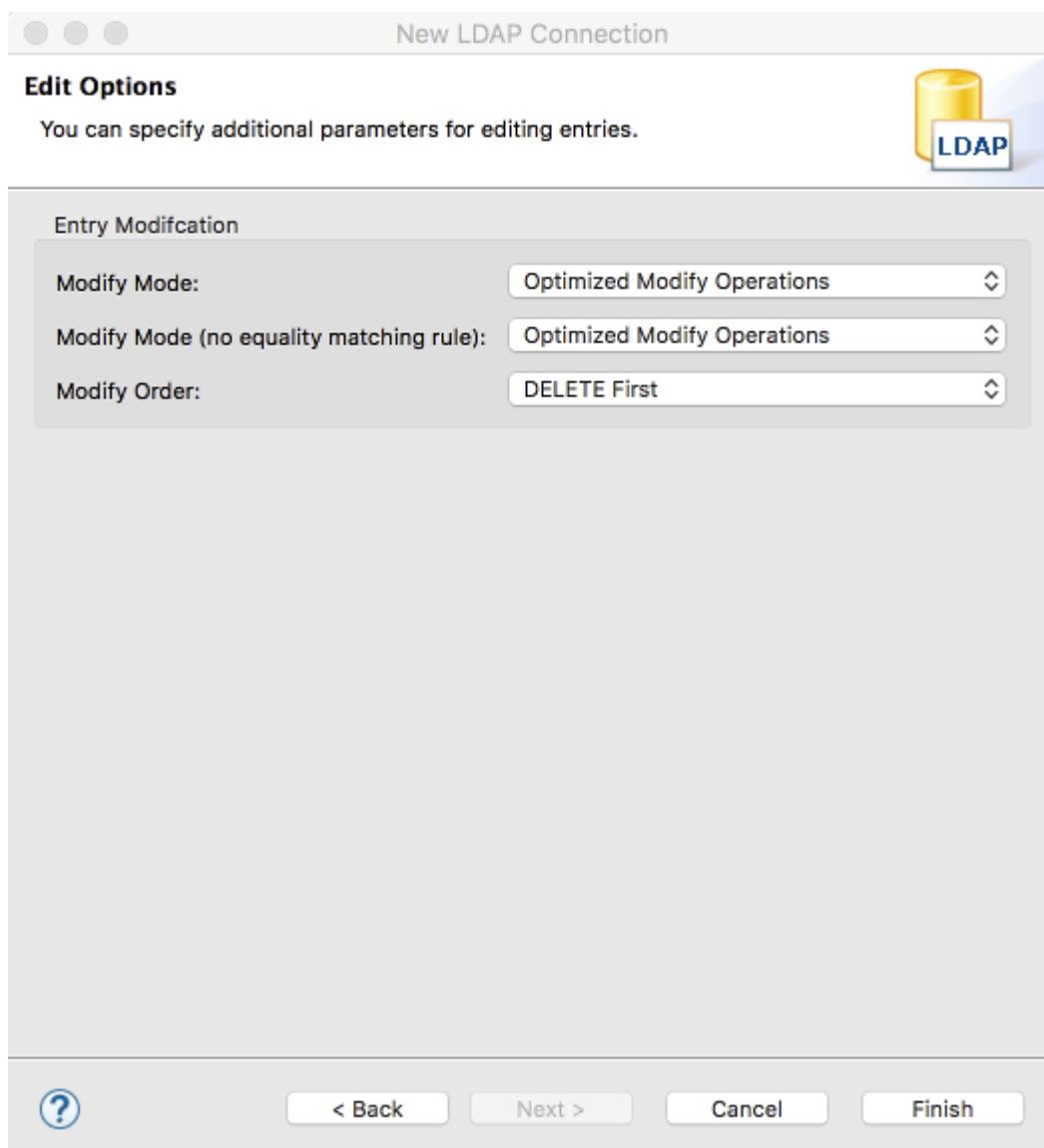
Endgültige Verbindung

Sie können die nächste Seite verlassen, wie es ist, und klicken Sie auf Fertig stellen.



Suche nach LDAP Baum

Der LDAP-Browser wird nun bevölkert und Sie können das LDAP-Verzeichnis durchsuchen.



Für die KNIME/Tomcat LDAP-Konfiguration erforderliche Informationen festlegen

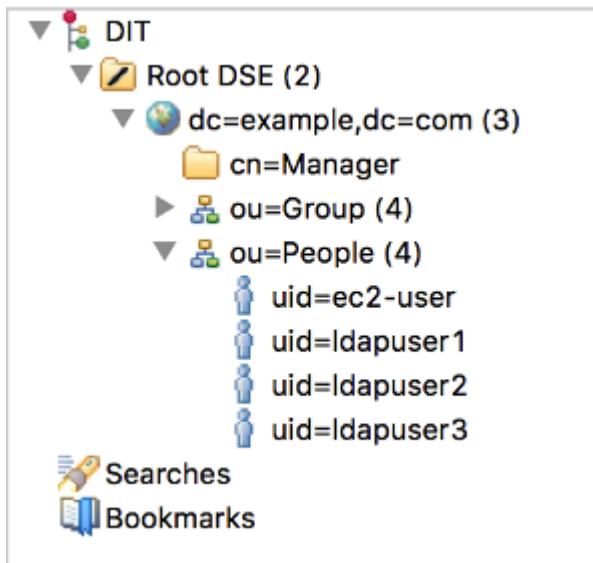
Erster Hinweis auf die [Tomcat Dokumentation über LDAP](#). Die Dokumentation ist sehr umfassend, wir destillierten einige der wichtigsten Punkte unten. Weitere Informationen finden Sie im Tomcat Dokumentation.

Im Grunde müssen wir etwas konstruieren, das aussieht:

```
UMWELT           "altep://52.50.222.127:389"
UserPattern = TOBEDETERMINed
RolleBase = TOBEDETERMINIEREN
RolleName = TOBEDETERMINIEREN
RolleSearch = TOBEDETERMINIEREN
>
```

Wir kennen die KontaktURL , da dies erforderlich war, um Apache Verzeichnis Studio.

Als nächstes müssen wir die Benutzer BasisEigentum. Der erste Artikel im Baum ist in der Regel der Basis DN, die die Benutzer BasisEigentum.



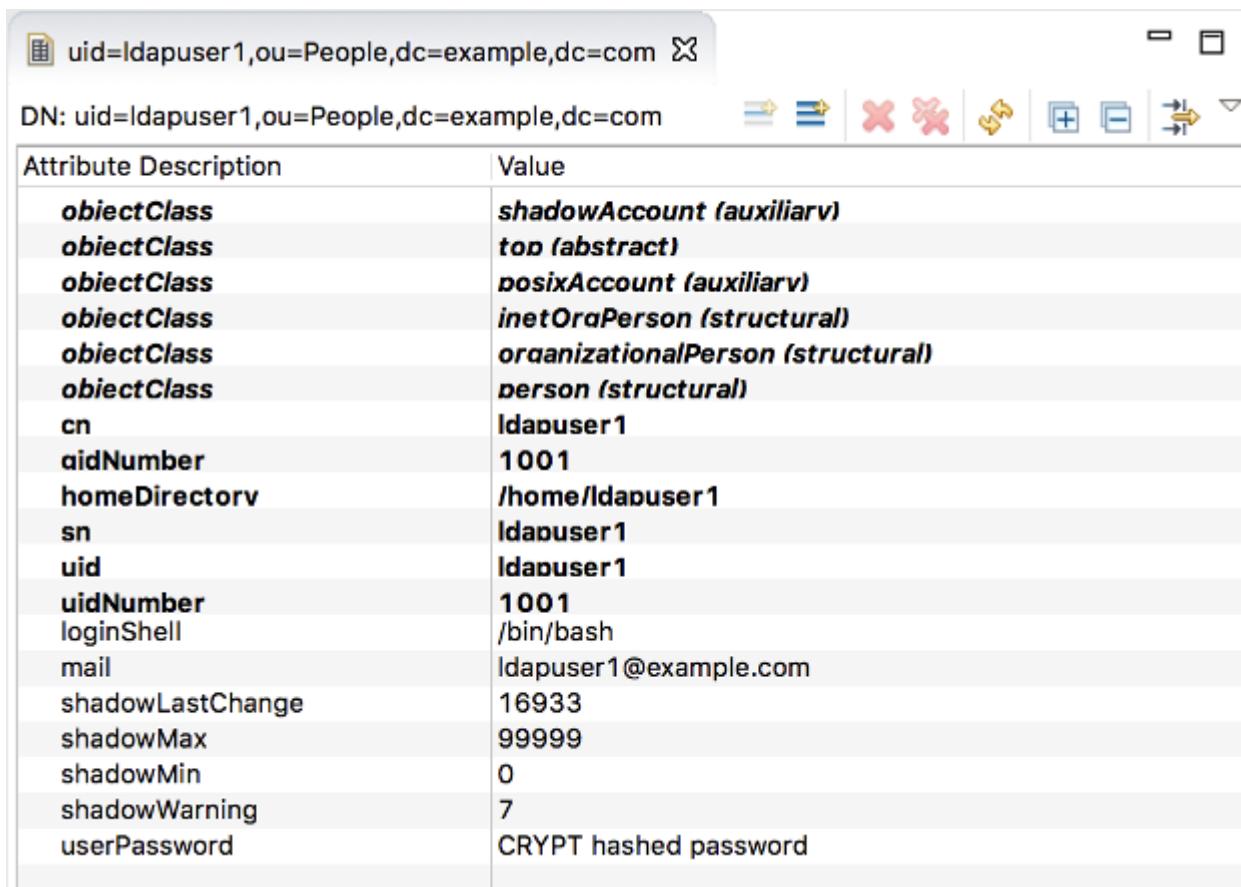
Sie können den Baum durchsuchen, um die Benutzer zu finden. In unserem Fall ou=People . Erweiterung des Unterbaums zeigt die Liste der Benutzer. In unserem Fall gibt es vier Benutzer ( ec2-User , Ldapuser1 , Ldapuser2 , Ldapuser3 )

Bestimmen Sie, ob Benutzer im Bind-Modus überprüft werden, oder Vergleichsmodus

Bind Mode

In unserem Fall, wenn sich Benutzer wie z. Ldapuser1 (der Benutzername ist derselbe wie der Schlüssel).

Wir kennen bereits die Basis DN und betrachten die Benutzerinformationen, die wir sehen, dass dieUid ist die Benutzername, die wir zur Authentifizierung verwenden möchten. So können wir die UserPattern .



The screenshot shows the KNIME Advanced Setup interface. At the top, it displays the DN: uid=ldapuser1,ou=People,dc=example,dc=com. Below this is a table with columns 'Attribute Description' and 'Value'. The table lists various attributes and their corresponding values for the user object.

Attribute Description	Value
<b>objectClass</b>	<b>shadowAccount (auxiliary)</b>
<b>objectClass</b>	<b>top (abstract)</b>
<b>objectClass</b>	<b>posixAccount (auxiliary)</b>
<b>objectClass</b>	<b>inetOrgPerson (structural)</b>
<b>objectClass</b>	<b>organizationalPerson (structural)</b>
<b>objectClass</b>	<b>person (structural)</b>
<b>cn</b>	<b>ldapuser1</b>
<b>aidNumber</b>	<b>1001</b>
<b>homeDirectory</b>	<b>/home/ldapuser1</b>
<b>sn</b>	<b>ldapuser1</b>
<b>uid</b>	<b>ldapuser1</b>
<b>uidNumber</b>	<b>1001</b>
<b>loginShell</b>	<b>/bin/bash</b>
<b>mail</b>	<b>ldapuser1@example.com</b>
<b>shadowLastChange</b>	<b>16933</b>
<b>shadowMax</b>	<b>99999</b>
<b>shadowMin</b>	<b>0</b>
<b>shadowWarning</b>	<b>7</b>
<b>userPassword</b>	<b>CRYPT hashed password</b>

Verwenden Sie die UserPattern : uid = {0},ou = people,dc = example,dc = com

So würde das Beispiel aussehen wie:

```
VerbindungURL = "ldap://52.50.222.127:389"
UserPattern = "uid={0},ou=people,dc=example,dc=com"
RolleBase = TOBEDETERMINIEREN
RolleName = TOBEDETERMIN
RolleSearch = TOBEDETERMINIEREN
>
```

Beachten Sie, dass wir noch nicht wissen, wie man Rolle Basis , RolleName , RolleSearch . Wir kommen zurück das später.

Vergleich

In diesem Fall gibt es keine one-to-one-Mapping zwischen dem Anmeldenamen und dem Benutzernamen, wir z.B. die E-Mail-Adressenkategorie verwenden möchten. In diesem Beispiel ist

Ldapuser1@example.com

The screenshot shows a table-based interface for editing LDAP attributes. The title bar says "uid=ldapuser1,ou=People,dc=example,dc=com". The table has two columns: "Attribute Description" and "Value".

Attribute Description	Value
<b>objectClass</b>	<b>shadowAccount (auxiliary)</b>
<b>objectClass</b>	<b>top (abstract)</b>
<b>objectClass</b>	<b>posixAccount (auxiliary)</b>
<b>objectClass</b>	<b>inetOrgPerson (structural)</b>
<b>objectClass</b>	<b>organizationalPerson (structural)</b>
<b>objectClass</b>	<b>person (structural)</b>
<b>cn</b>	<b>ldapuser1</b>
<b>aidNumber</b>	<b>1001</b>
<b>homeDirectory</b>	<b>/home/ldapuser1</b>
<b>sn</b>	<b>ldapuser1</b>
<b>uid</b>	<b>ldapuser1</b>
<b>uidNumber</b>	<b>1001</b>
<b>loginShell</b>	<b>/bin/bash</b>
<b>mail</b>	<b>ldapuser1@example.com</b>
<b>shadowLastChange</b>	<b>16933</b>
<b>shadowMax</b>	<b>99999</b>
<b>shadowMin</b>	<b>0</b>
<b>shadowWarning</b>	<b>7</b>
<b>userPassword</b>	<b>CRYPT hashed password</b>

Um diese Art von Anmeldung durchzuführen, benötigen wir einen Vergleichsmodus:

Hier ist die Basis DN für Benutzer Basis, und wir müssen auch definieren UserSearch . Hier sind wir werden suchen Postsendung.

```
Name: "cn=Manager,dc=example,dc=com"
KontaktPassword = "Geheimnis"
VerbindungURL = "ldap://52.50.222.127:389"
Benutzer Basis "ou=people,dc=example,dc=com"
UserSearch = "(mail={0})"
BenutzerRoleName = "memberOf"
RolleBase = TOBEDETERMINIEREN
RolleName = TOBEDETERMINIEREN
RolleSearch = TOBEDETERMINIEREN
>
```

Zugang zur Gruppe

Nun, da Benutzer authentifiziert sind, müssen wir die Gruppen konfigurieren, die Zugriff haben:

Dafür brauchen wir Rolle Basis und RolleName Parameter. Sie können die ou=Gruppe für weitere Informationen. Hier nehmen wir das Beispiel, dass die hrpeople Gruppe

sollte auf KNIME Server zugreifen können.

Attribute Description	Value
<b>objectClass</b>	<b><i>groupOfNames (structural)</i></b>
<b>cn</b>	<b>hrpeople</b>
<b>member</b>	<b>cn=ldapuser3,ou=people,dc=example,dc=com</b>
<b>member</b>	<b>cn=ldapuser2,ou=people,dc=example,dc=com</b>
<b>description</b>	<b>Human Resources group</b>

Im Beispiel ist der Wert Mitglied, nach dem wir suchen wollen, 'Mitglied'.

Was zur Konfiguration führt:

```
VerbindungURL = "ldap://52.50.222.127:389"
UserBase = "ou=people,dc=example,dc=com"
UserSearch = "(mail={0})"
BenutzerRoleName = "memberOf"
RolleBase =
        "ou=Group,dc=example,dc=com"
(Name)
        "cn"
RolleSearch =
        "(member={0})"
>
```

Es gibt eine zweite Möglichkeit, bei der Gruppenmitgliedschaft in den Benutzerdaten gespeichert wird (dies ist ungewöhnlich, und nicht in diesem Führer abgedeckt. Siehe die vollständige Tomcat-Dokumentation).

Auch eingebettete Rollen (wo eine Rolle/Gruppe andere Rollen/Gruppen enthalten kann) sind möglich, in denen

Das ist der **RolleNestes** Parameter. Z. Gruppe 'IT', enthält einige Benutzernamen, plus

'Windows', 'UNIX', 'Mac' Gruppen. Diese Gruppen können auch Untergruppen enthalten.

Hoffentlich haben Sie jetzt die Details, die Sie benötigen, um KNIME Server mit LDAP zu verbinden.

### Active Directory Beispiel

Wenn Sie Active Directory als Benutzerdatenbank verwenden und an der Standardstruktur festhalten,

Die folgende Konfiguration dient als guter Ausgangspunkt:

```
Verbindung Name = "cn=Manager,dc=example,dc=com"
KontaktPasswort = "geheim"
KontaktURL = "ldap://52.50.222.127:389"
UserSubtree = "true"
Benutzer Basis= "cn=Benutzer,dc=domain,dc=com"
UserSearch = "(sAMAccountName={0})"
BenutzerRoleName = "memberOf"
Rolle Basis = "cn=Benutzer,dc=domain,dc=com"
RolleName = "cn"
roleSearch = "(member={0})"
RolleSubtree = "true"
RolleNested = "true"/>
```

Sie müssen die drei hervorgehobenen Verbindungsparameter sowie die beiden anpassen

c) Werte

in der **Benutzer Basis** und **Rolle Basis**. Die anderen Parameter können in der Regel verwendet werden, wie sie sind.

### Combined Realm

Es ist möglich, einen kombinierten Bereich einzurichten, in dem sowohl die Benutzerdatenbank als auch LDAP

Authentifizierung wird parallel verwendet. Generell wird dies nicht empfohlen, kann aber nützlich sein für

debugging und initial setup/testing. Das folgende Beispiel zeigt, wie das funktionieren könnte.

```
RessourceName = "UserDatabase"/>

VerbindungURL = "ldap://52.50.222.127:389"
UserBase = "ou = people,dc = example,dc = com"
UserSearch = "(mail = {0})"
BenutzerRoleName = "memberOf"
RolleBase = "ou = Gruppe,dc = Beispiel,dc = com"
RolleName = "cn"
roleSearch = "(member = {0})"/>
```

## Verschlüsselte LDAP

Falls Sie eine verschlüsselte LDAP-Authentifizierung verwenden und Ihr LDAP-Server eine Selbst-unterzeichnetes Zertifikat, Tomcat wird es ablehnen. In diesem Fall müssen Sie den LDAP-Server hinzufügen Zertifikat an die globale Java-KeyStore, die in Verzeichnis/lib/security/cacerts :

```
Schlüsseltool -Import -v -noprompt -trustcacerts -file
-keystore /lib/security/cacerts -storepass changeit
```

Alternativ können Sie die Ccerts Datei, fügen Sie Ihr Serverzertifikat hinzu und fügen Sie Folgendes hinzuzwei Systemeigenschaften /conf/catalina.properties :

```
javax.net.ssl.trustStore =
javax.net.ssl.keyStorePasswort = changeit
```

## Fehlerbehebung

In einigen Fällen möchten Sie zusätzliche Log-Dateiinformationen über den LDAP extrahieren Authentifizierungsprozess. In diesem Fall können Sie bearbeiten Apache...
tomcat\*/conf/logging.properties zu ergänzen:

```
org.apache.catalina.realm.level = ALL
org.apache.catalina.realm.useParentHandlers = true
org.apache.catalina.authenticator.level = ALL
org.apache.catalina.authenticator.useParentHandlers = true
```

Sobald Sie die Änderungen vorgenommen haben, müssen Sie den knime-server Prozess/Service neu starten.

Wenn Sie Ihr Problem erfolgreich debugged, vergessen Sie nicht, kommentieren oder entfernen  
 diese Zeilen von der Loggen. Eigenschaften Datei, wie es unnötig große Protokolldateien erstellen.

## Konfigurieren von Single-Sign-On mit Kerberos und LDAP

Single-Sign-On kann für KNIME Server konfiguriert werden. Dazu gehören das WebPortal, aber auch alle andere Dienstleistungen (REST, SOAP, etc.) KNIME Server bietet.

Die hierfür verwendete Technologie ist Kerberos, ein Netzwerkprotokoll, das für Authentifizierung durch Tickets und starke Verschlüsselung. Im folgenden wird davon ausgegangen dass Sie mit den Grundkonzepten von Kerberos und LDAP vertraut sind, wie im vor. Sie finden umfassende Dokumentation für die neueste Version von Kerberos [Hier](#).

Dieser Abschnitt beschreibt Schritt für Schritt, wie man die Kerberos-Authentifizierung mittels einer **Aktives Verzeichnis Service** und **Windows Clients**. Andere Setups sind möglich und können erfordern verschiedene Verfahren, die funktionsfähig sind.

 Die meisten Setups werden in bestimmten Aspekten von diesem Leitfaden abweichen, so machen gegebenenfalls Anpassungen.

Kerberos erfordert Setup für alle drei beteiligten Parteien: den Kerberos- und LDAP-Service (Active Verzeichnis), der Tomcat Server mit KNIME Server und die Clients.

### Active Directory Konfiguration

Der erste Schritt ist, das Active Directory korrekt einzurichten. Es wird angenommen, dass Sie bereits eine Active Directory-Domain mit Benutzern und korrekten Gruppen für die KNIME Server-Nutzung eingerichtet. Zusätzliche Schritte speziell für Kerberos sind:

ANHANG Erstellen Sie einen technischen Benutzer für den Tomcat Server in LDAP.

2. Associate a Service Principal Name (SPN) mit dem neu erstellten Benutzer für die Tomcat Server. Öffnen Sie dazu eine Windows PowerShell und geben Sie Folgendes ein:

Setspn - HTTP/TOMCAT\_FQDN@REALM TECHNISCHE/USER

Im obigen Befehl ersetzen

TOMCAT\_FQD mit dem vollqualifizierten Domainnamen (FQDN) der Maschine, die läuft KNIME Server (und damit der Tomcat-Server),

- REALEN mit dem Kerberos-Bereich Ihrer Active Directory-Installation,
- und TECHNISCHE/USER mit dem Namen des technischen Benutzers, den Sie im vorheriger Schritt.

Es ist wichtig, daß TOMCAT\_FQD die DNS-Einträge (FQDN zu IP) sowie DNS-Einträge (IP to FQDN) können durch den Domänencontroller behoben werden und alle Kunden.

### 3. Stellen Sie sicher, dass die richtigen Verschlüsselungsmethoden auf dem Domänencontroller aktiv sind:

The screenshot shows the Windows Local Security Policy snap-in. The left pane displays a tree structure under 'Local Policies' with 'Security Options' selected. The right pane lists various policies with their current status. The policy 'Network security: Configure encryption types allowed for Kerberos' is highlighted, showing 'Not Defined'.

Policy	Security Setting
Network access: Sharing and security model for local accounts	Classic - local
Network security: Allow Local System to use computer identity for ...	Not Defined
Network security: Allow LocalSystem NULL session fallback	Not Defined
Network security: Allow PKU2U authentication requests to this com...	Not Defined
<b>Network security: Configure encryption types allowed for Kerberos</b>	<b>Not Defined</b>
Network security: Do not store LAN Manager hash value on next pa...	Enabled
Network security: Force logoff when logon hours expire	Disabled
Network security: LAN Manager authentication level	Not Defined
Network security: LDAP client signing requirements	Negotiate si...
Network security: Minimum session security for NTLM SSP based (i...	Require 128-

- a. Gehen Sie Verwaltungstools → Lokale Sicherheitspolitik
- B. Durchsuchen Sicherheitseinstellungen/Lokale Richtlinien/Sicherheitsoptionen
- c. Der Eintrag finden Netzwerksicherheit: Verschlüsselungstypen für Kerberos konfigurieren . wenn der Wert nicht definiert wird, dann sind alle Verschlüsselungstypen erlaubt. Wenn es definiert ist, machen sicher, dass es mindestens die Methoden enthält: RC4\_HMAC , AES128 , AES256 und Zukunft Verschlüsselungsarten .

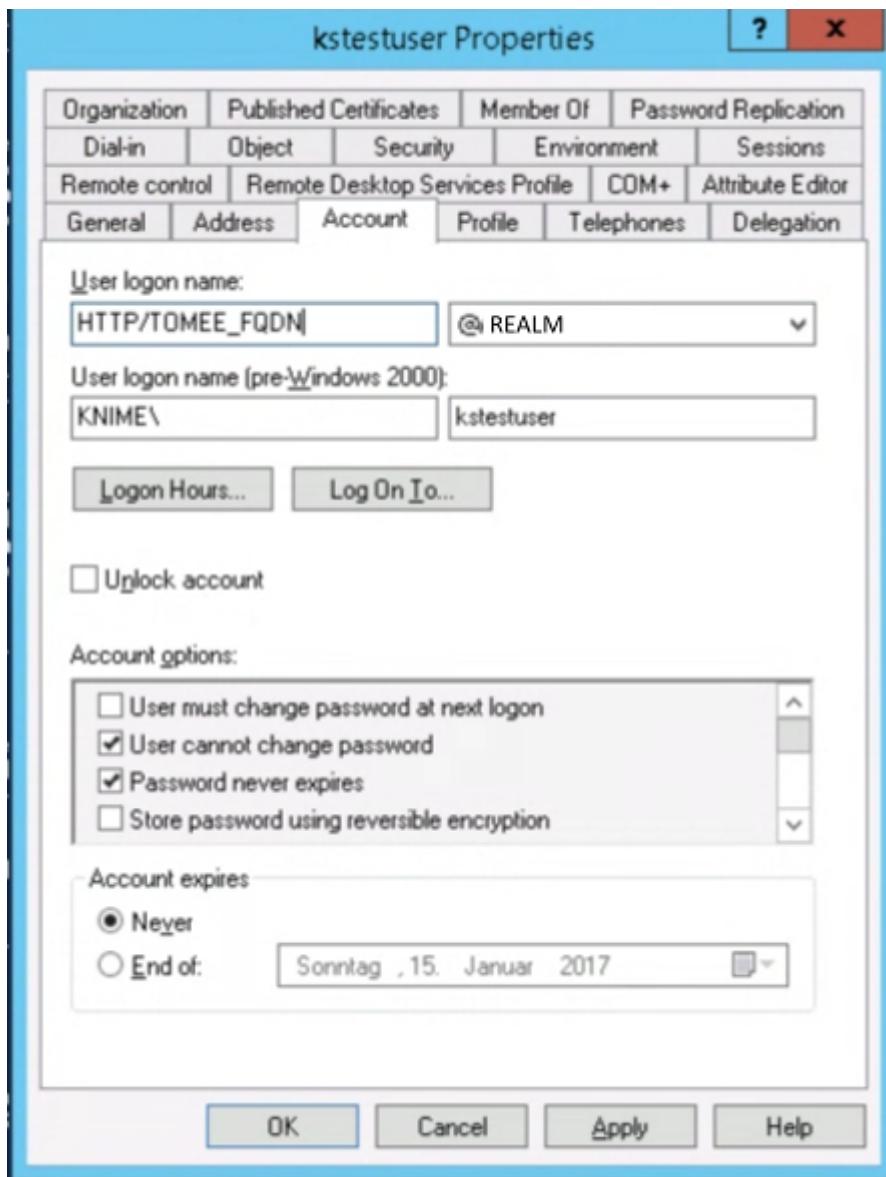
### 4. Öffnen Sie eine Windows-Power Shell und erstellen Sie eine Schlüssel Datei mit dem folgenden Befehl.

Einstellen der Werte nach Ihren Einstellungen:

```
ktpass /out PATH/tomcat.keytab
/mapuser TECHNICAL_USER@REALM
/princ HTTP/TOMCAT_FQDN@REALM
/Pass +rndPass
/crypto AES256-SHA1 Typ KRB5_NT_PRINCIPAL
```

Die erstellte Keytab-Datei muss später auf den Tomcat Server kopiert werden.

### 5. Öffnen Sie die "Benutzereigenschaften" in Active Directory für den technischen Tomcat Benutzer, den Sie haben erstellt. Gehen Sie dann auf die Registerkarte "Konto" und stellen Sie sicher, dass die folgenden Einstellungen festgelegt sind:



- a. Benutzer-Logo-Name ist richtig gesetzt
- B. Passwort läuft nie ab = wahr
- c. Benutzer kann Passwort nicht ändern = wahr
- d. Dieses Konto unterstützt Kerberos AES 128 Bit Verschlüsselung = wahr
- e. Dieses Konto unterstützt Kerberos AES 256 Bit Verschlüsselung = wahr
- f. Verwenden Sie Kerberos DES Verschlüsselung für dieses Konto = falsch (empfohlen)
6. Dann gehen Sie auf die "Delegation" Tab und setzen Sie die Radio-Taste, um zu jedem Service (nur Kerberos) Vertraue diesem Benutzer für die Delegation

## Tomcat Server Konfiguration

ANHANG KNIME installieren Server wie in der

[KNIME Installationsanleitung für Server](#)

2. Stellen Sie geeignete Konfigurationseinstellungen wie in der

[KNIME Server Verwaltungshandbuch](#)

3. LDAP-Authentifizierung in der

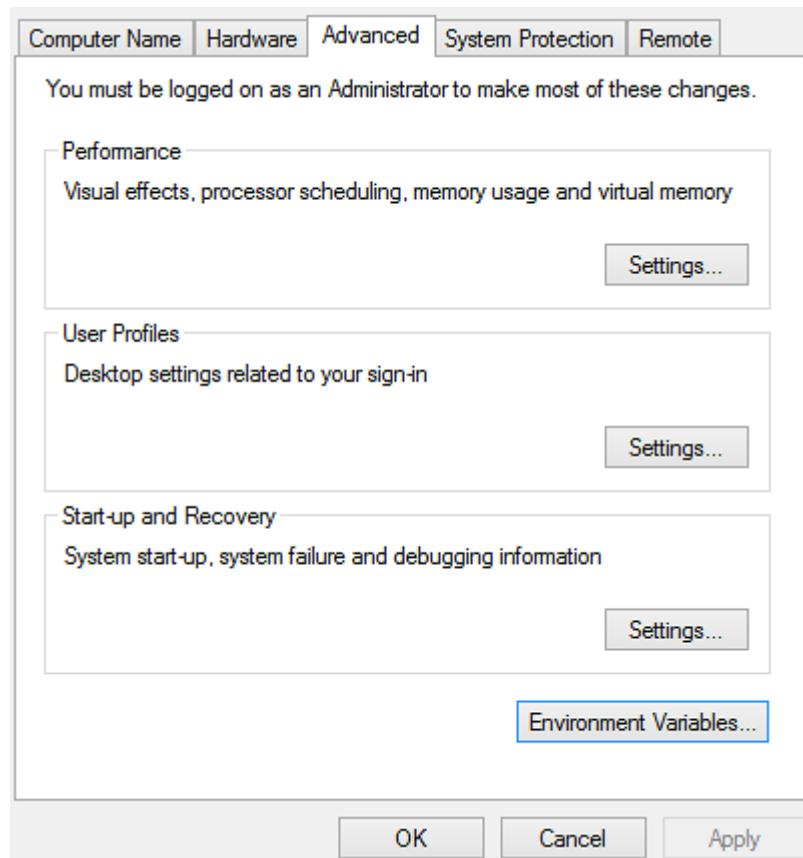
[Server.xml](#) die Verbindung zu Ihrem Active Directory, wie  
die Verbindung zu Ihrem Active Directory, wie  
. Beachten Sie, dass es

notwendig, um einen temporären Listing-Benutzer zu erstellen, um die LDAP-Lookups durchzuführen. Dieser Schritt ist optional, aber empfohlen, zu testen, dass die grundlegende LDAP-Authentifizierung funktionsfähig ist.

L 347 vom 20.12.2013, S. 1). Überprüfen Sie, ob die Umgebungsvariablen JAVA\_HOME und CATALINA\_HOME werden richtig definiert:

- JAVA\_HOME sollte auf das JDK-Home-Verzeichnis (enthaltend a Bin Ordner)
- CATALINA\_HOME sollte auf das Tomcat-Verzeichnis hinweisen (enthaltend a Bin Ordner).

Unter Windows kann dies in Systemsteuerung → System → Erweiterte Systemeinstellungen



a. Klicken Sie auf Umweltvariablen

B. In der Systemvariables Gruppe überprüfen, ob

JAVA\_HOME und

CATALINA\_HOME . Erstellen oder Einstellen der Werte entsprechend.

Auf Linux erstellen oder ändern Sie die Werte in

/etc/default/knime-Server

5. Sobald ein Arbeitsstandard LDAP-Setup überprüft wurde, stellen Sie eine Sicherung der Inhalte

von /conf durch Kopieren an /conf\_ldap

6. Unter Windows, geöffnet Nachdruck und folgendes tun:

a. Navigieren

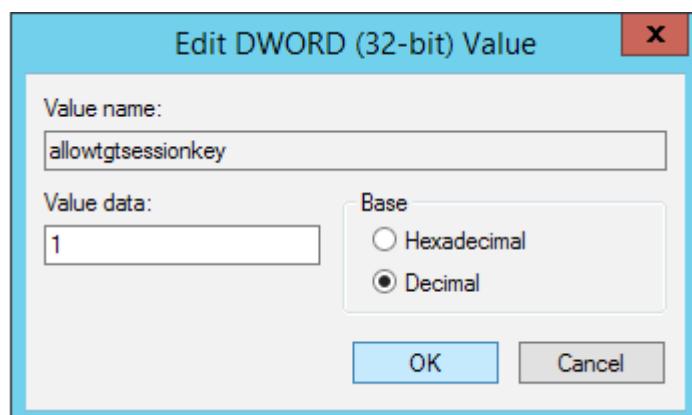
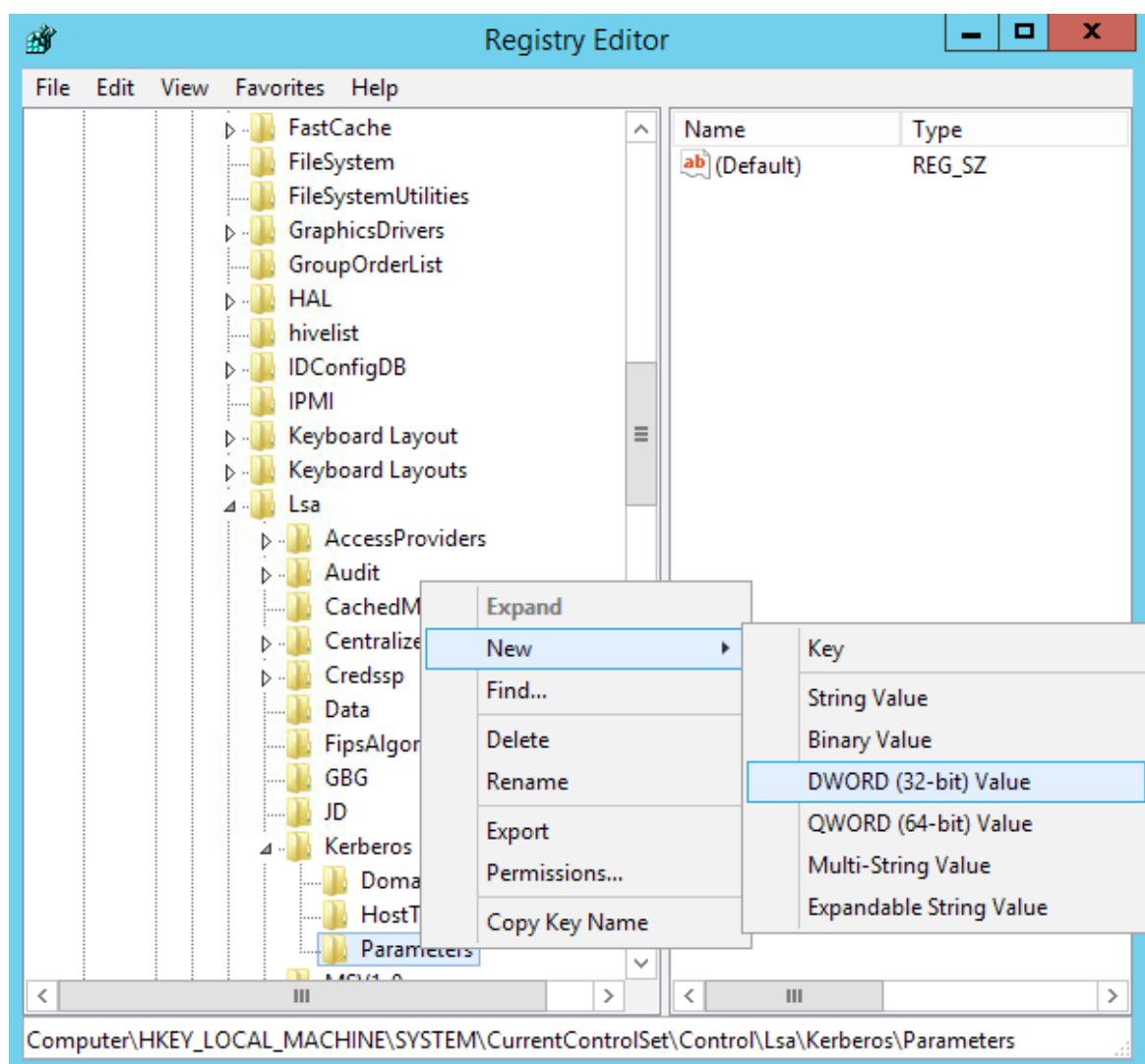
HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters

Er .

B. Schlüssel hinzufügen Fazit

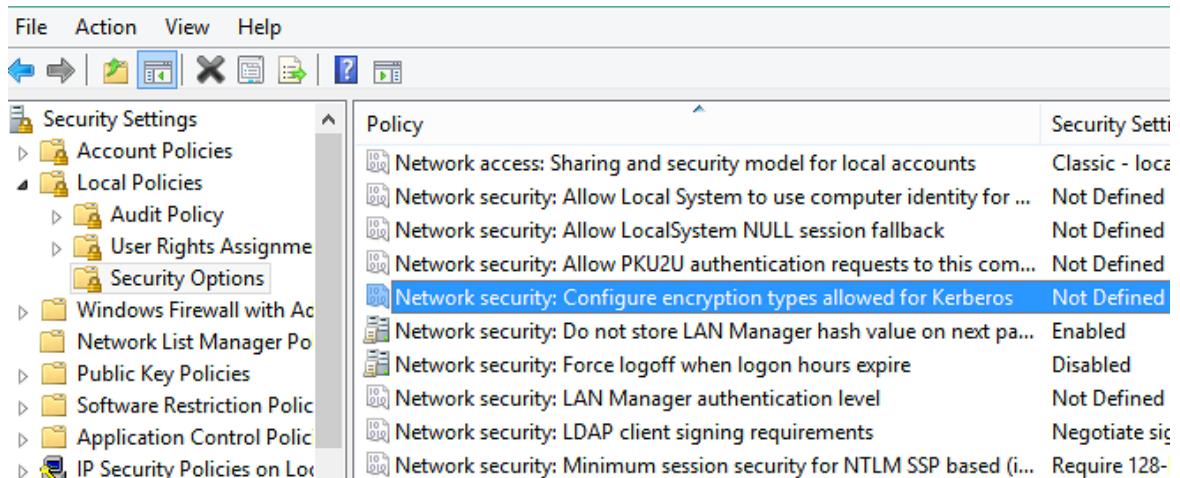
( REG\_DWORD ) und den Wert auf

1.



7. Achten Sie unter Windows darauf, dass die richtigen Verschlüsselungsmethoden für Kerberos aktiv sind:

- a. Gehen Sie Verwaltungstools → Lokale Sicherheitspolitik
- B. Durchsuchen Sicherheitseinstellungen/Lokale Richtlinien/Sicherheitsoptionen
- c. Der Eintrag finden Netzwerksicherheit: Verschlüsselungstypen für Kerberos konfigurieren



- d. Wenn der Wert nicht definiert ist, sind alle Verschlüsselungstypen erlaubt. Wenn es definiert ist, machen sicher, dass es mindestens die Methoden enthält: RC4\_HMAC, AES128, AES256 und Zukunft Verschlüsselungsarten.

8. Kopieren Sie die zuvor erstellte Keytab-Datei für die SPN an einen Ort Ihrer Wahl.

Empfohlen würde /conf/

ANHANG Erstellen einer krb5.ini Datei in /conf/. Der Inhalt der Datei sollte aussehen

wie:

```
(libdefaults)
default_realm = REALM
default_keytab_name = "FILE:/conf/tomcat.keytab"
default_tkt_enctypes = aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96
default_tgs_enctypes = aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96
weiterföhrend = true

(Realms)
REALM { \cHFFFF }

kdc = DOMAIN_CONTROLLER_FQDN:88
}

[domain_realm]
Ihrdomain.com = REALM
*.yourdomain.com = REALM
```

Einstellen der Werte nach Ihrer Konfiguration, aber halten Sie die FILE: Präfix für die keytab Name. Wenn Sie einen anderen Standort oder Dateinamen für diese Datei verwenden möchten, können Sie

so durch die Definition der folgenden Java-Systemeigenschaft in  
 /conf/system.properties :

```
java.security.krb5.conf=PATH_TO_KRB5_CONF
```

#### 10. Erstellen oder Bearbeiten der Datei conf/jaas.conf

. Der Inhalt der Datei sollte

wie:

```
com.sun.security.jgss.krb5.accept {\cHFFFF}
com.sun.security.auth.module.Krb5LoginModule
erforderlich
Nicht verfügbar
Haupt = "HTTP/TOMCAT_FQDN@REALM"
keyTab = "/conf/tomcat.keytab"
Speichern Sie die Datei
VerwendungKeyTab = true
NutzungTicketCache = true
isInitiator = true
aktualisierenKrb5Config = true
ModulBanner = true
storePass = true;
};

com.sun.security.jgss.krb5.initiativ {\cHFFFF}
com.sun.security.auth.module.Krb5LoginModule
erforderlich
Nicht verfügbar
Haupt = "HTTP/TOMCAT_FQDN@REALM"
keyTab = "/conf/tomcat.keytab"
Speichern Sie die Datei
VerwendungKeyTab = true
NutzungTicketCache = true
isInitiator = true
aktualisierenKrb5Config = true
ModulBanner = true
storePass = true;
};
```

Passen Sie die Werte entsprechend Ihrer Konfiguration an. Beachten Sie, dass der Ort zum Keytab  
 Datei muss als absoluter Pfad angegeben werden und enthalten vorwärts Slashes, auch unter Windows.

Wenn Sie einen anderen Standort oder Dateinamen für die  
 Definition der folgenden Java-Systemeigenschaft in  
 /conf/system.properties :

Jaas.conf du kannst das

```
java.security.auth.login.conf=PATH_TO_JAAS_CONF
```

In der Kerberos-Dokumentation wird diese Datei oft als

Anmeldung

11. Fügen Sie die folgende Eigenschaft auf die Liste der JVM-Systemeigenschaften beim Start hinzu. Sie können definiert in /conf/system.properties :

```
javax.security.auth.useSubjectCredsOnly = false
```

12. Konfigurieren der KNIMEServerAutorisierung Ventil:

a. Navigieren /conf/Catalina/localhost/

B. Bearbeiten Sie die ~~Knime.xml~~ Datei (der Name der Datei ist gleich dem Kontext root, der war im KNIME Server-Installer eingestellt, der Standard ist Knospen, wenn die ~~knime.war~~ Datei war umbenannt in ~~umbenannt.war~~, die xml-Datei wird aufgerufen ~~umbenannt.xml~~)

c. Die Linie finden

```
Insgesamt
className = "com.knime.enterprise.tomcat.authenticator.KnimeServerAuthenticator
" EnableSpnego = "false"
basicAuthPaths = "/rest,/webservices" formAuthPaths = "/" />
```

d. Ändern Sie es

```
Insgesamt
className = "com.knime.enterprise.tomcat.authenticator.KnimeServerAuthenticator
" EnableSpnego = "true"
basicAuthPaths = "/rest,/webservices" />
```

e. Standardmäßig werden die REST und SOAP Webservices eingerichtet, um grundlegende HTTP zu verwenden Authentifizierung. Wenn Sie Single-Sign-On auch für den REST und/oder SOAP verwenden möchten Webservices, z.B. wenn Sie einen REST-Client verwenden, der Kerberos unterstützt, den BasicAuthPaths das Attribut entsprechend. Es ist eine komma getrennte Liste von Pfaden Überschreiben der Standard-Authentifizierungsmethode. Das Attribut löschen ermöglicht Kerberos für alle Dienstleistungen.

Zum Beispiel, wenn REST mit Single-Sign-On das Attribut verwendet werden soll würde so aussehen: basicAuthPaths = "/webservices"

13. Ändern der Server.xml und die JNDIREalm Einstellungen zur Verbindung mit Ihrem LDAP. wenn Sie haben Ihr Setup in Schritt 3 erfolgreich getestet, es genügt, die Verbindung Name und KontaktPasswort Attribute.

Bitte beachten Sie, dass bei Kerberos die Verbindung Name und KontaktPasswort Attribute

werden ignoriert. Auch die Verwendung der UserPattern wird von Tomcat nicht unterstützt  
Kerberos. Verwendung Benutzer Basis in Kombination mit UserSearch statt.

Die Realm-Definition könnte so aussehen:

```
VerbindungURL = "ldap://dc.domain.com:3268"
UserSubtree = "true"
UserBase = "cn = Benutzer,dc = domain,dc = com"
UserSearch = "(sAMAccountName = {0})"
BenutzerRoleName = "memberOf"
RolleBase = "cn = Benutzer,dc = domain,dc = com"
RolleName = "cn"
roleSearch = "(member = {0})"
RolleSubtree = "true"
RolleNested = "true" />
```

Wenn Sie Kerberos in einem kombinierten Bereich verwenden, stellen Sie sicher, dass das JNDIRealm mit  
Ihr LDAP ist **erste** in der Liste der Reiche.

#### 14. KNIME Server für die Änderungen. Logfiles inspizieren

/logs um sicherzustellen, dass es keine Fehlermeldungen über Ihre  
Änderungen.

#### Konfiguration des Clients

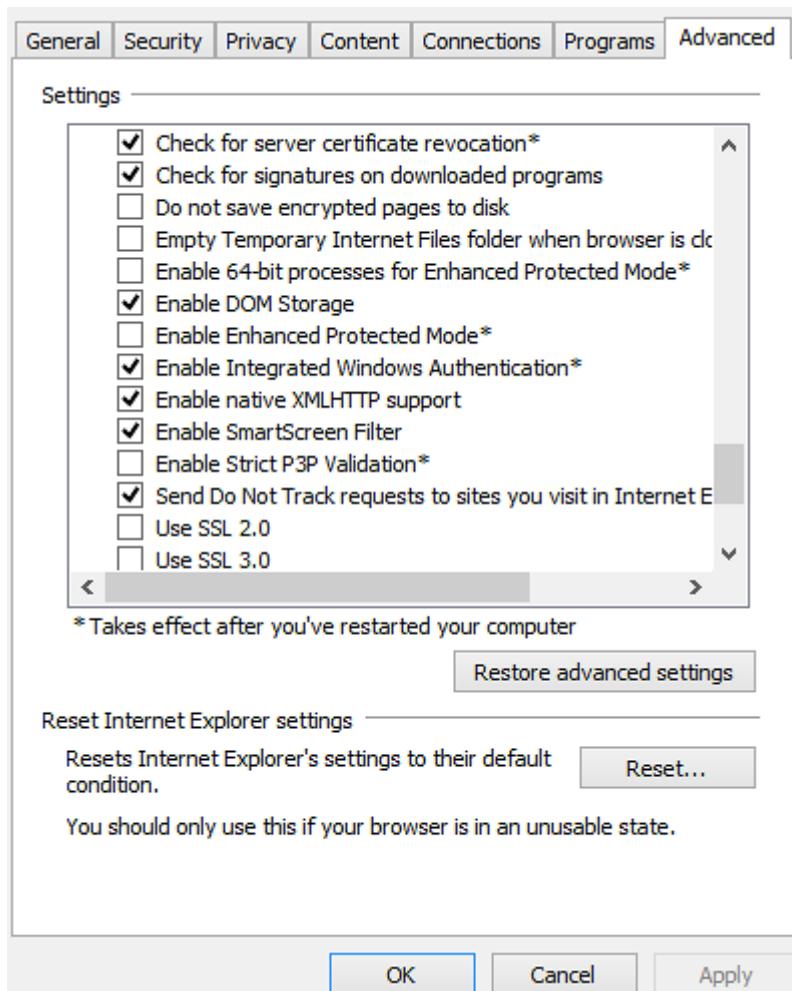
Client-Konfiguration erfordert zwei Schritte:

**ANHANG** Die Client-Maschine muss Teil der Domain sein, und der Endbenutzer eingeloggt darin  
Domain.

2. Alle vom Client verwendeten Browser müssen die Kerberos-Authentifizierung aktiviert haben. Die  
folgende Abschnitte beschreiben, wie man Internet Explorer und Firefox für Kerberos  
Authentifizierung.

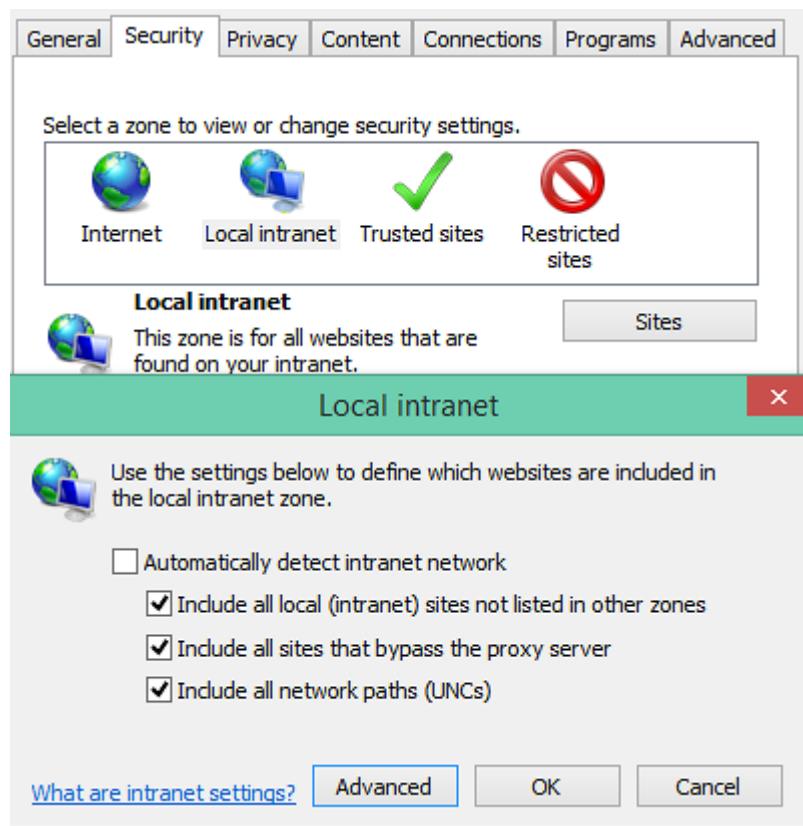
Kerberos Authentication im Internet Explorer aktivieren

**ANHANG** Öffnen Sie das Menü "Internetoptionen" und durchsuchen Sie die Registerkarte "Erweitert".



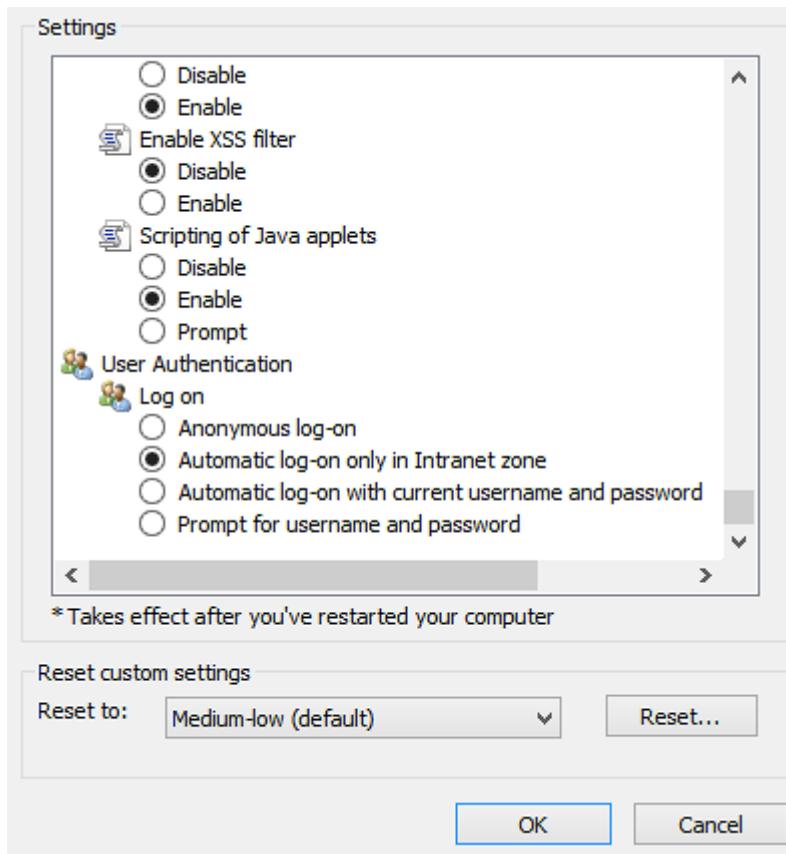
Die Einstellung "Enable Integrated Windows Authentication" muss überprüft werden.

2. Durchsuchen Sie die Registerkarte "Sicherheit", wählen Sie "Local Intranet" und klicken Sie auf die Schaltfläche "Sites".



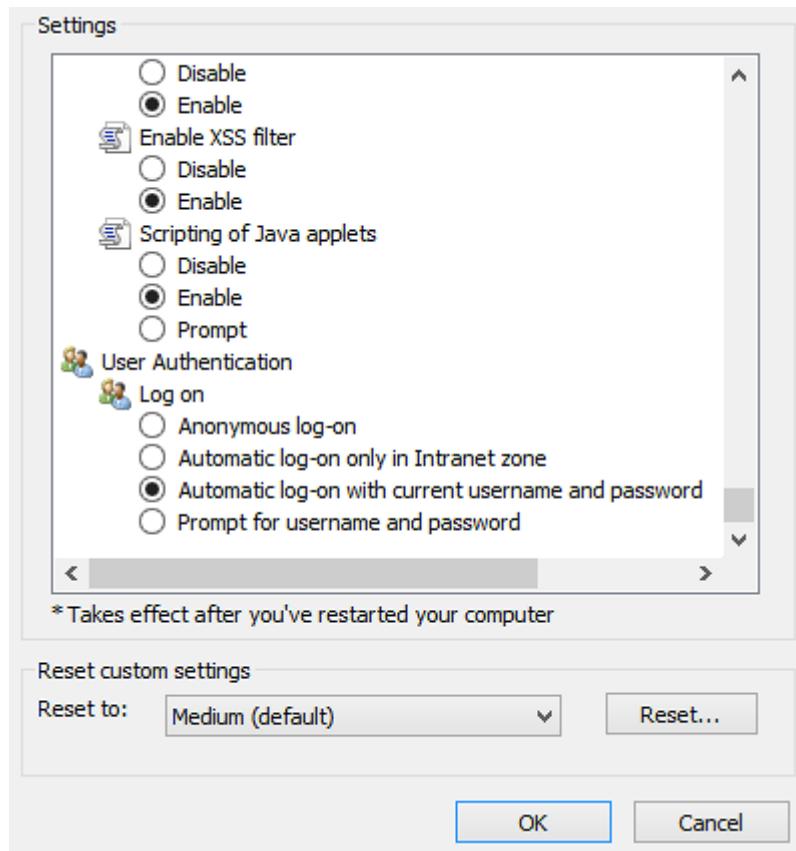
3. Klicken Sie auf "Erweitert" und fügen Sie die URL von KNIME Server in die Liste der Websites in der Zone.

4. Klicken Sie auf "Custom Level" und überprüfen Sie das in Lokale Intranet Sicherheitsstufe → Benutzer Authentifizierung wird auf "Automatische Anmeldung nur in Intranet-Zone" gesetzt



5. Es könnte notwendig sein, auch KNIME Server in die Liste der vertrauenswürdigen Seiten hinzuzufügen. Um das zu tun, geh. auf "Trusted Sites" klicken und auf die Schaltfläche "Sites" klicken. Die URL von KNIME Server in die Liste der Websites in der Zone.

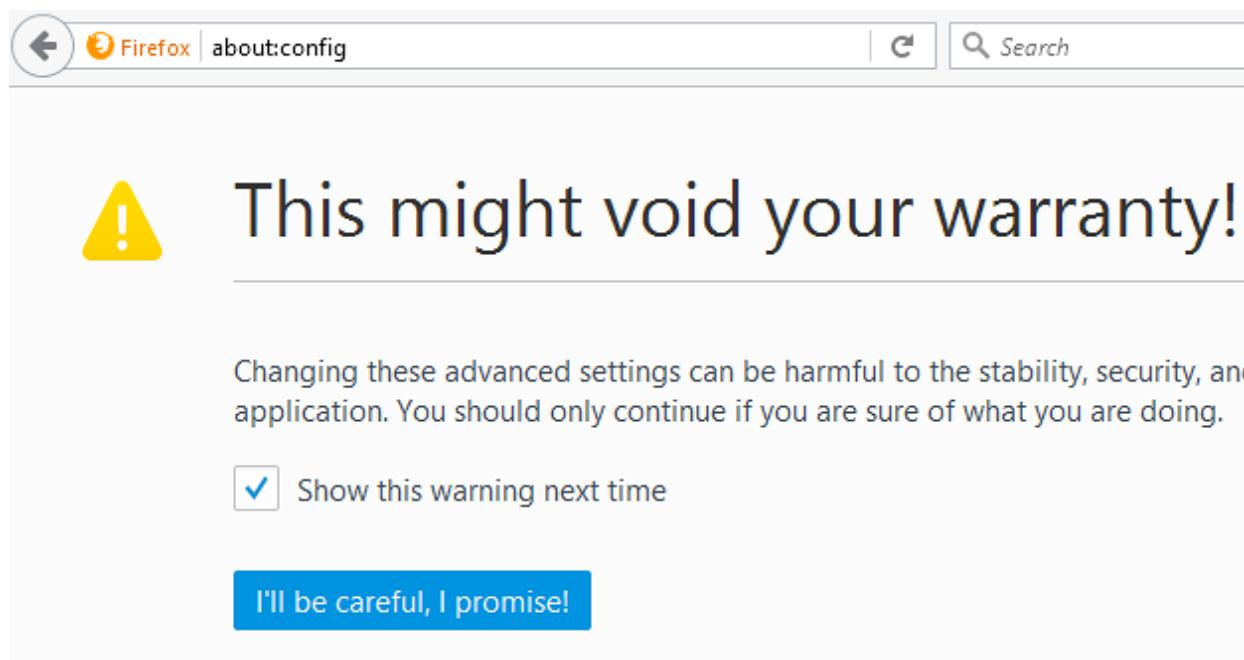
6. Überprüfen Sie, ob Trusted Sites Sicherheitsstufe → Benutzeroauthentifizierung wird auf "Automatic" gesetzt Anmeldung mit aktuellem Benutzernamen und Passwort".



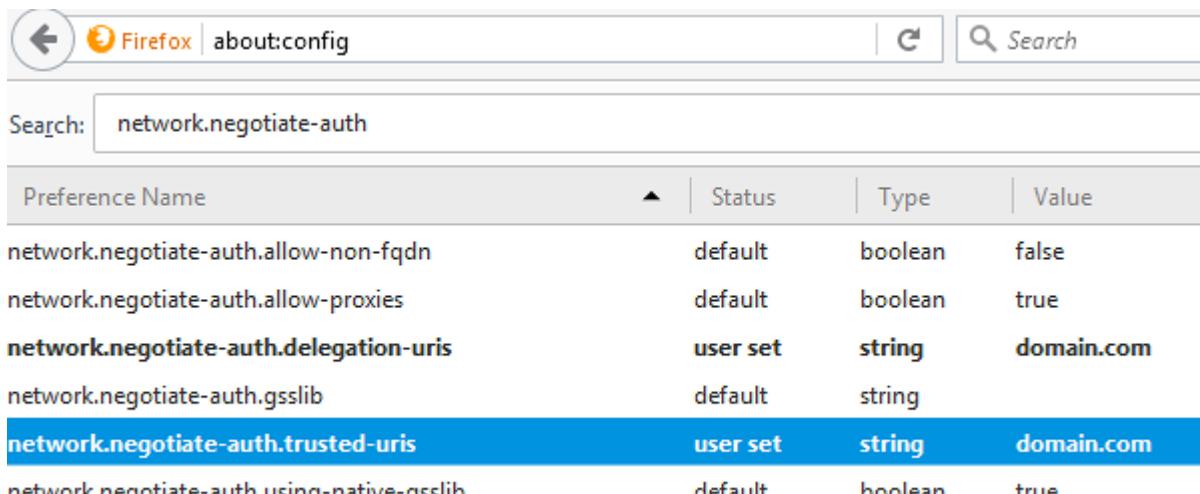
#### Kerberos Authentication in Firefox aktivieren

ANHANG Starten Sie Firefox und Typ `about:config` in der Adressleiste.

2. Ignorieren Sie die Warnung, indem Sie auf die "Ich werde vorsichtig sein, ich verspreche!" Taste.



3. Finden Sie die entsprechenden Einstellungen durch Eingabe `Netzwerk.negotiate-auth` im Suchfeld.



Preference Name	Status	Type	Value
network.negotiate-auth.allow-non-fqdn	default	boolean	false
network.negotiate-auth.allow-proxies	default	boolean	true
<b>network.negotiate-auth.delegation-uris</b>	<b>user set</b>	<b>string</b>	<b>domain.com</b>
network.negotiate-auth.gsslib	default	string	
<b>network.negotiate-auth.trusted-uris</b>	<b>user set</b>	<b>string</b>	<b>domain.com</b>
network.negotiate-auth.using-native-gsslib	default	boolean	true

Änderung der network.negotiate-auth.delegation-uris und Network.negotiate-auth.trusted-uris die URL von KNIME Server enthalten. Es könnte genug sein, nur einzugeben Ihre Domain.

## Fehlerbehebung

Ein Kerberos-Setup ist in der Regel sehr komplex und benötigt präzise Konfiguration. Fehlermeldungen sind oft kryptisch. Um einen Kerberos-Setup zu debuggen, ist es sehr hilfreich, zusätzliche Logging für die Authentifizierung im Tomcat Server. Dazu können Sie einige Dinge konfigurieren.

ANHANG Um die Protokollierung in den Krb5-Modulen zu ermöglichen, fügen Sie oder aktivieren Sie die folgenden beiden Zeilen in beiden Abschnitt der Jaas.conf (oder Anmeldung in /conf :

```
debug=true
ModulBanner=true
```

Beachten Sie, dass der Debug-Ausgang nur auf die Konsole gedruckt wird.

2. Um die Debug-Ausgabe der Kerberos-Implementierung in Java zu erhöhen, fügen Sie folgende hinzu:

System-Eigenschaft auf Start (kann in system.properties Datei in /conf :)

```
-Dsun.security.krb5.debug=true
```

3. Debugging für Authentifizierungs- und Realm-Module hinzufügen, indem

Loggen. Eigenschaften Datei in /conf . Für Klarheit alle Authentifizierungsausgabe kann in einer separaten Datei eingeloggt werden.

[...]

```
4auth.org.apache.juli.FileHandler.level = FINE
4auth.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
4auth.org.apache.juli.FileHandler.prefix = auth.

[...]

org.apache.catalina.realm.level = ALL
org.apache.catalina.realm.handler = 4auth.org.apache.juli.FileHandler

org.apache.catalina.authenticator.level = ALL
org.apache.catalina.authenticator.handlers = 4auth.org.apache.juli.FileHandler

com.knime.enterprise.tomcat.handlers = 4auth.org.apache.juli.FileHandler
com.knime.enterprise.tomcat.level = DEBUG

org.apache.juli.logging.UserDataHelper.CONFIG = INFO_ALL

org.apache.coyote.http11.level = DEBUG
org.apache.coyote.http11.handlers = 4auth.org.apache.juli.FileHandler
```

# Dynamische Profile für Server-Managed Anpassungen

Wie in der [KNIME Leitfaden für die Verwaltung von Servern](#) es ist möglich, einen benutzerdefinierten zu schreiben einen Profilanbieter, der den Server und die Liste der Profile dynamisch auswählt. Diese benutzerdefinierte Anbieter muss eine Umsetzung der `org.knime.product.profile.IProfileProvider` Schnittstelle, die in der `org.knime.product` Einsticken. Die Umsetzung dieser Schnittstelle darf keine Klassen verwenden, die Leseinstellungen auslösen, andernfalls die Standardeinstellungen können nicht mehr geändert werden. Dies beinhaltet die Verwendung von allgemein verwendeten KNIME-Klassen wie `KNIMECon` oder `NodeLogger`.

Daher schlagen wir vor, ein neues Plug-in zu erstellen, das nur eine Abhängigkeit von `org.knime.product` (für die `IProfileProvider` Schnittstelle) und verwendet keine anderen KNIME andere Klassen. Eine Ausnahme sind Klassen von `org.knime.core.util` Einsticken, weil es verwendet keine Vorlieben (und wird nie sein). Andere als dies, die Umsetzung ist Geradeaus. Die Klasse `org.knime.product.profile.ExampleProfileProvider` enthält ein minimales Beispiel eines benutzerdefinierten Profilanbieters, den Sie als Ausgangspunkt verwenden können. Nicht vergessen Sie, Ihre Implementierung am Erweiterungspunkt zu registrieren `org.knime.product.profileProvider`.

Wenn Sie Fragen zur Implementierung eines benutzerdefinierten Profilanbieters haben, wenden Sie sich bitte an Wir.

# OpenID Connect Authentication

Das KNIME Server kann konfiguriert werden, um einen OpenID Connect aktivierten Identitätsanbieter für Authentifizierung.

Wenn die Funktion aktiviert ist und ein authentifizierter Benutzer konfiguriert wird, wird auf eine KNIME Server-Benutzer.

 Bitte beachten Sie, dass nur der Authorization Code Flow unterstützt wird und nicht die ID Token Flow.

## Authentication Valve Konfiguration für OpenID Connect (OAuth)

Um die OAuth-Authentifizierung für den KNIME Server zu aktivieren, die Datei  
tomcat>/conf/Catalina/localhost/knime.xml muss bearbeitet werden.

Es ist möglich, den Server so zu konfigurieren, dass sowohl die OAuth- als auch die Grundauthentifizierung mithilfe Anmeldeinformationen können gleichzeitig verwendet werden. Für die Konfiguration die folgenden Parameter kann der Definition des Authentifizierungsventils hinzugefügt werden:

**EnableOAuth = ""**

Ermöglicht die OAuth-Authentifizierung.  
Der Standard ist falsch (OAuth-Authentifizierung deaktiviert).

**aktivierenBasicAuthWithOAuth = ""**

Ermöglicht die grundlegende Authentifizierung entlang der Seite OAuth, wenn OAuth aktiviert ist. Der Standard ist falsch (basische Authentifizierung deaktiviert bei Verwendung von OAuth). Mit dieser Option aktiviert, REST-Clients und die KNIME Analytics-Plattform können weiterhin mit dem Nutzer authentifizieren Anmeldeinformationen (Benutzername und Passwort). Einloggen im WebPortal ist nur möglich OAuth über den Identity Provider verwenden.

**oAuthConfiguration Pfad = ""**

Der Pfad zur Konfigurationsdatei.  
Die empfohlene Lage ist /conf/Catalina/localhost/knime-  
oidc-config.json

Der Ventileintrag sollte ähnlich aussehen wie folgt:

```
aktivierenSpnego = "false" basicAuthPaths = "/rest" formAuthPaths = "/"
secretKey = "someSecretKey" aktivierenOAuth = "true" aktivierenBasicAuthWithOAuth = "false"
oAuthConfigurationPath = "/path/to/conf/Catalina/localhost/knime-oidc-config.json"/>
```

Die anbieterspezifische Konfiguration erfolgt durch Erstellen einer

knime-oidc-config.json

Datei und

Einsetzen nach dem in

tomcat>/conf/Catalina/localhost/knime.xml

, wo

tomcat>/conf/Catalina/localhost/knime-oidc-config.json

ist der empfohlene Ort.

Hier ist ein Beispiel für eine solche Datei, die Parameter werden unten erläutert:

```
"Identity-provider-name": "Some Identity Provider",
"auth-server-url": "https://identity.provider/",
"Ressource": "client-id",
"Erklärungen": {\cHFFFF}
"Geheimnis": "Client-secret"
},
"Zusatz-Zulassungs-Endpunkt-Parameter": "Zusatz-Parameter=some-
Wert&some-other-parameter=some-value",
"Zusatzoskope": "Zusatzoskop ein anderes Mikroskop",
"principal-attribute": "claim-used-for-principal-mapping"
}
```

Bitte beachten Sie, dass dies ein

JSON Datei, so sollte es die

JSON Standard,

z.B. Komma nach jedem Eintrag außer dem letzten.

Der Authentiker ist in der Lage, die notwendigen OpenID Connect Endpunkte zu entdecken

automatisch. Es tut dies, indem man auth-server-url und das ".well-

bekannt/offenid-configuration" Endpunkt, wenn es verfügbar ist. Wenn die Entdeckung die Endpunkte ausfällt

muss explizit eingestellt werden.

"identity-provider-name": ""

Der Name des Identitätsanbieters, der auf der Anmeldelandingpage angezeigt wird, als:

"Login mit "

"auth-server-url": ""

Die Basis-URL für den OpenID Connect Endpoint des Identity Providers.

Dieser Endpunkt wird für die automatische Endpunkt-Erkennung verwendet, nach

[Angemeldet bleiben](#)

[Entdeckung](#).

**"Autorisierungsendpunkt": ""**

(OPTIONAL) Der Autorisierungsendpunkt des Identity Providers.

Muss explizit festgelegt werden, wenn die automatische Endpunkt-Erkennung ausfällt.

**"token-endpoint": ""**

(OPTIONAL) Der Tokenendpunkt des Identity Providers.

Muss explizit festgelegt werden, wenn die automatische Endpunkt-Erkennung ausfällt.

**"jwks-endpoint"**

(OPTIONAL) Der JWKS Endpunkt, um JWT Tokens zu überprüfen.

Muss explizit festgelegt werden, wenn die automatische Endpunkt-Erkennung ausfällt.

**"userinfo-endpoint": ""**

(OPTIONAL) Der Endpunkt Benutzerinfo des Identity Providers.

Muss explizit festgelegt werden, wenn die automatische Endpunkt-Erkennung ausfällt.

**"Ressource": ""**

Die Client-ID der Anwendung.

Ja

**"Ergebnisse": {"secret": ""}**

(OPTIONAL) Das Client-Geheimnis, wenn es für den Identity Provider gesetzt werden muss.

Ja

**"öffentlich-client": ""**

(OPTIONAL) Wenn keine Client-Anmeldeinformationen benötigt werden, setzen Sie sich auf „wahr“.

Ja

**"Zusätzliche Genehmigungsendpoint-Parameter": "&"**

(OPTIONAL) Zusätzliche Parameter, die beim Aufruf des Identitätsanbieters verwendet werden autorisierung endpoint.

Ja

**"Zusatzoskope": " "**

Raumgetrennte Liste zusätzlicher Bereiche, die beim Aufruf der Identity Provider Berechtigung Endpoint. Die mit dem Berechtigungsendpoint. geöffnet Anwendungsbereich immer

**"principal-attribute": ""**

Die Behauptung, die verwendet werden sollte, um die authentifizierten Benutzer auf den tomcat Bereich zu mappen. Diese Forderung wird auch als Kontoname für die Benutzer verwendet, die auf das KNIME zugreifen Server. Standardmäßig wird dies Teil Anspruch. Die Spitzname Anspruch könnte für stellen Sie in diesem Fall sicher, dass der entsprechende Geltungsbereich angefordert wird, wenn den Berechtigungsendpoint anrufen.

**"Minimal-Zugang-Token-Pasing": ""**

Wenn der Zugriff auf Token nicht überprüft werden kann, weil einer seiner Ansprüche fehlerhaft ist oder nicht der Spezifikation entsprechen (siehe <https://openid.net/specs/openid-connect-kern-1.0.html>), der Zugang zu den Zeichen auf ein Minimum gehalten werden kann, so daß die tokens Unterschrift und Emittent können noch überprüft werden.

**"allow-opaque-access-token": ""**

Wenn der Identity Provider keinen Zugriff auf Token als JWT bieten kann, kann der opake Token nicht durch Überprüfung ihrer Ansprüche oder durch Überprüfung ihrer Unterschrift überprüft werden. Die Überprüfung ist dann links zum Identitätsanbieter, indem Sie den userinfo Endpoint. Das ist nicht empfohlen, aber es ist der einzige Weg, um Kompatibilität für Identitätsanbieter zu ermöglichen, dass keine JWT-Zugriffstoken zur Verfügung stellen.

**"treat-access-token-as-opaque": "**

Wenn der Zugriff auf die Signatur des Tokens nicht überprüft werden kann, weil er nicht dem Spezifikation, zum Beispiel, weil es Header enthält, die behandelt werden müssen besonders zur Unterschriftenprüfung können die Token als undurchsichtig behandelt werden. In diesem Fall die Zugriffstoken werden nicht parsiert und deren Unterschrift und Emittent nicht überprüft. Die Überprüfung wird dann dem Identitätsanbieter überlassen, indem der userinfo Endpoint aufgerufen wird. Das wird nicht empfohlen, aber kann als Workaround für unvereinbare Identität helfen Anbieter, die nicht der OIDC Spezifikation entsprechen. Wenn diese Option auf "wahr" gesetzt wird, sowohl die minimal-access-token-parsing, als auch die zulassen-opaque-access-token Option wird ignoriert.

"principal-attribute-to-username-regex": ""

(OPTIONAL) Der Hauptbeitrag kann mit einem regelmäßigen Ausdruck geändert werden. Für Beispiel, wenn die E-Mail-Anforderung als Principal-attribute konfiguriert ist, eine E-Mail wie Pressemitteilungen könnte abgebildet werden john.do , mit dem Regex "@company.com". Der erste Teil, der dem Hauptbeitrag entspricht ein regelmäßiger Ausdruck wird entfernt.

"Redirect-rewrite-rules": "

(OPTIONAL) Könnte benötigt werden, gibt die Redirect URI Umschreiben Regeln an. Das ist ein Objektnotation, der Schlüssel ist ein regelmäßiger Ausdruck, dem die Redirect URI sein soll angepasst und der Wert ist der Ersatz String.

"perform-direct-redirect": ""

(OPTIONAL) Wenn diese Option auf "wahr" gesetzt wird, wird die KNIME Webportal Landingpage nicht gezeigt werden. Stattdessen wird der Benutzer direkt an den Identitätsanbieter weitergeleitet für Authentifizierung. Der Benutzer wird noch mit der Landung präsentiert, wenn der Login ausfällt.

"proxy-url": "> proxy-url >"

(OPTIONAL) Die Proxy-Konfiguration, die verwendet werden sollte, um mit der Identität zu sprechen Provider.

Der Konfigurationsparameter in Form von  
http://user:password@hostname:port .

## Benutzer- und Gruppenmanagement

Wenn die Funktion aktiviert ist, werden die Benutzer mit Hilfe eines konfigurierbarer Anspruch vom Userinfo Endpoint. Zum Beispiel, wenn der Anspruch Spitzname gewählt wird, Lassen Sie es "john.doe" sein, der Benutzer wird einem internen Benutzer mit dem Benutzernamen "john.doe" zugeordnet.

## Einschränkung von Login-Gruppen

Standardmäßig können sich alle authentifizierten Benutzer in den KNIME Server einloggen. Zugriff sollte eingeschränkt werden, erlaubt Login-Gruppen können in der [KNIME Serverkonfiguration Datei](#), so können sich Benutzer, die keiner Gruppe zugeordnet sind, nicht an den Server anmelden.

Fügen Sie die folgende Konfigurationsoption hinzu, wird das Login auf die angegebenen Gruppen einschränken:

```
com.knime.server.login.allowed_groups = > group >,, ...
```

Definiert die Gruppen, die zum Server einloggen dürfen.

Standardwert ermöglicht Benutzern aller Gruppen.

## Tomcat Group Management

Standardmäßig wird das Gruppenmanagement durch den tomcat-Bereich bearbeitet. Hier entlang des Tomcat realms können noch entsprechend der [Benutzerauthentifizierung](#) Abschnitt in der [KNIME Serververwaltung Leitfaden](#). Beachten Sie, dass das Passwort für den Benutzer im tomcat realm gesetzt wird spielt keine Rolle für die OAuth-Authentifizierung. Solange der definierte Anspruch mit Benutzernamen in der Datenbank wird der Benutzer mit den zugewiesenen Gruppen eingeloggt.

## LDAP Konfiguration

Die Verwendung des JNDIRealm für LDAP ist auch eine gültige Konfiguration, solange der Hauptbeitrag die den Benutzernamen abbildet, kann auf einen Eintrag in LDAP abgebildet werden. Eine gültige Konfiguration für Benutzer/Gruppen-Lookup, mit der Haupt-Attribute-E-Mail, könnte so aussehen:

```
RessourceName = "UserDatabase" />

VerbindungURL = "ldap://ldap.hostname:389"
userBase = "ou=people, dc=company, dc=com"
userSearch = "(email={0})"
roleBase = "ou=groups,dc=company,dc=com" roleName = "cn"
roleSearch = "(member={0})" />
```

[Bitte beachten Sie die KNIME Server Advanced Setup Guide](#) für weitere Informationen zur Einrichtung LDAP.

## Gruppen-Mapping-Anforderung

Der Authentiker kann auch konfiguriert werden, um eine benutzerdefinierte Forderung aus der userinfo abrufen Endpunkt zu Benutzergruppen. Dies könnte beispielsweise ein "Gruppen" genannter Anspruch sein. Die Gruppe Ein Retrieval kann durch Definition des Anspruchs im knime-oidc-config.json ermöglicht werden, mit Parameter:

**Gruppen-Mapping-claim = ""**

Der Name des Anspruchs, der eine Reihe von Gruppen für den Benutzer auf der userinfo zur Verfügung stellt Endpoint. Stellen Sie sicher, dass die Reichweiten entsprechend definiert werden, so dass der Anspruch abgerufen werden.

Wenn der Gruppen-Mapping-Antrag aktiviert ist, wird der tomcat-Bereich nicht verwendet.

**Azure Active Directory und Graph API**

Um die Gruppenverwaltung über das Azure Active-Verzeichnis zu aktivieren, ist der KNIME Server Authentisierung kann konfiguriert werden, um die Microsoft Graph API für Gruppenabruf. Die Die folgende Tabelle beschreibt die verfügbaren Konfigurationsparameter, die in der knime-oidc-config.json.

**graph-api-group-information = ""**

Wenn dies auf "wahr" gesetzt wird, wird der Authentifikator die Graph-API rufen, um den Benutzer abzurufen Gruppen.

Ja

**graph-api-use-display-name = ""**

Wenn dies auf "wahr" gesetzt wird, wird der displayName der Gruppe als Gruppenname verwendet, wenn er verfügbar. Wenn der DisplayName nicht abrufbar ist, wird die Gruppen-ID als Gruppe verwendet Name. Wenn der Parameter auf "false" gesetzt wird, wird die Gruppe id in jedem Fall verwendet.

Es ist erforderlich, dass die Anwendung das Verzeichnis hat. Lesen.ALL API Erlaubnis mit admin Einverständnis, damit die Gruppen über die Graph API abgerufen werden können.

**KNIME Server Client-Konfiguration für die KNIME Analytics Platform**

Um die OAuth-Authentifizierung für Clients eines für OpenID konfigurierten KNIME Servers zu aktivieren Verbinden Sie, einige zusätzliche Konfiguration muss dem repository>/config/knime-server.config Datei.

Standardmäßig werden die Endpunkte, Client-ID und Client-Geheimnisse mit dem knime-oidc- konfiguriert config.json.

Die folgende zeigen, wie die zusätzliche Konfiguration aussehen könnte, die Konfiguration Parameter werden nachfolgend beschrieben:

```
com.knime.server.authentication.oauth.redirect_ports = 8888,8881,8882
com.knime.server.authentication.types = OAuth,Credentials
com.knime.server.authentication.preferredType = OAuth
com.knime.server.authentication.oauth.token_refresh_rate = 5m
com.knime.server.authentication.oauth.allowed_clock_skew = 30s
```

**com.knime.server.authentication.types = >type>, >type>**

Comma-getrennte Liste von Authentifizierungstypen. Mögliche Werte sind "OAuth" und "Credentials". Dies sollte die Konfiguration des Authentifizierungsventils in tomcat>/conf/Catalina/localhost/knime.xml

**com.knime.server.authentication.preferredTyp = >type>**

Der bevorzugte Authentifizierungstyp, der von der KNIME Analytics Platform verwendet werden sollte.

Der bevorzugte Typ wird beim Hinzufügen von KNIME Server als neue Halterung vorgewählt.

Punkt in der KNIME Analytics Platform. Mögliche Werte sind "OAuth" und "Credentials".

Dies sollte den in

com.knime.server.authentication.type

**com.knime.server.authentication.oauth.redirect\_ports = „>port>**

Eine komma-separierte Liste von Häfen, die für die Autorisierung Umleitung verwendet werden sollten. wenn keine Liste wird ein zufälliger Port für jede Berechtigung gewählt. Einige Identität Anbieter können Wildcards nicht unterstützen, in diesem Fall sollte eine Liste der konfigurierten Ports hier vorgesehen sein.

**com.knime.server.authentication.oauth.token\_refresh\_rate =**

z.B. 5m, 30m oder 2h>

(OPTIONAL) Ist der Zugriffstoken opak, ist das Ablaufdatum des Zugriffstokens kann nicht ermittelt werden, so dass eine Erfrischungsrate eingeführt werden kann, um den Token zu erfrischen.

**com.knime.server.authentication.oauth.allowed\_clock\_skew =**

z.B. 30s oder 1m>

Die erlaubte Uhr, die bei der Parsierung des Zugriffs auf Token in KNIME Analytics verwendet wird Plattform. Dies kann verwendet werden, um potenziellen Takt zwischen der Identität zu handhaben Anbieter und KNIME Analytics Platform.

**com.knime.server.authentication.oauth.client\_secret=**

(OPTIONAL) Verwenden Sie diese Option nur, wenn sie vom Identitätsanbieter benötigt wird. Der Kunde geheim, das von der KNIME Analytics Platform verwendet werden muss, um gegen den Identitätsgeber.

**com.knime.server.authentication.oauth.authorization\_endpoint=**

**Endpunkt**

(OPTIONAL, geliefert aus knime-odic-config.json)

Der Autorisierungsendpunkt des Identity Providers.

**com.knime.server.authentication.oauth.token\_endpoint=**

(OPTIONAL, geliefert aus knime-odic-config.json)

Der Tokenendpunkt des Identity Providers.

**com.knime.server.authentication.oauth.client\_id=**

(OPTIONAL, geliefert aus knime-odic-config.json)

Die Client-ID der Anwendung.

**com.knime.server.authentication.oauth.scope = > scope > „, > scope >**

(OPTIONAL, geliefert aus knime-odic-config.json)

Komma-getrennte Liste von zusätzlichen Bereichen, die beim Aufruf der Identity Providers Berechtigung Endpoint. Die Anwendungsbereiche sollten den in den knime-oidc-config.json auf der Serverseite. Die **geöffnet** Umfang sollte in jedem Fall, zusammen mit den zusätzlichen Bereichen auch im knime-oidc-config.json definiert.

**com.knime.server.authentication.oauth.additional\_query\_params=**

(OPTIONAL, geliefert aus knime-odic-config.json)

Zusätzliche Parameter, die beim Aufruf des Identity Providers verwendet werden sollten als Abfrage-String vorgesehene Berechtigungsendpunkt.

Die Callback-URL für die KNIME Analytics Platform ist

```
http://127.0.0.1:/oauthredirectlistener
```

wobei von den konfigurierten Umleitungsports abhängt.

## Debugging OIDC Authentication

Um die Protokollierung zu ermöglichen, Authentifizierungsprobleme zu debuggen, müssen einige Änderungen

Anwendung der `/conf/logging.properties`

Erstens, der Henker **4auth.org.apache.juli.FileHandler** muss in der Liste der Griffe hinzugefügt werden:

```
Handler = 4auth.org.apache.juli.FileHandler,....
```

Zweitens sollte der log-Level für diesen Handler eingestellt werden

ALLGEMEIN:  
NE

```
4auth.org.apache.juli.FileHandler.level = ALL
4auth.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
4auth.org.apache.juli.FileHandler.prefix = auth.
4auth.org.apache.juli.FileHandler.maxDays = 90
```

Schließlich sollte der Block über die Authentifizierung nicht kommentiert werden:

```
#org.apache.catalina.realm.level = ALL
#org.apache.catalina.realm.handlers = 4auth.org.apache.juli.FileHandler
#org.apache.catalina.authenticator.level = ALL
#org.apache.catalina.authenticator.handlers = 4auth.org.apache.juli.FileHandler
com.knime.enterprise.tomcat.handlers = 4auth.org.apache.juli.FileHandler
com.knime.enterprise.tomcat.level = ALL
#org.apache.juli.logging.UserDataHelper.CONFIG = INFO_ALL
#org.apache.coyote.http11.level = DEBUG
#org.apache.coyote.http11.handlers = 4auth.org.apache.juli.FileHandler
```

Mit dieser Log-Konfiguration kann die Log-Ausgabe zur Authentifizierung unter  
`tomcat>/logs/auth.yyyy-mm-dd.log`

## Mit der eigenen Tomcat Installation

KNIME Server basiert auf Apache Tomcat daher ist es möglich, Ihren eigenen Apache zu verwenden

Tomcat Installation statt auf die mit KNIME Server verpackte Version zu verlassen. Dies kann

nützlich sein, wenn Sie eine bestimmte Version von Apache Tomcat benötigen

Das unterscheidet sich von dem, was der KNIME Server Installer bietet.

Während es technisch möglich ist, KNIME Server innerhalb eines bestehenden Apache Tomcat zu betreiben

Installation, die bereits andere Web-Anwendungen, die wir empfehlen, mit einem eigenen

Installation. Der Grund ist, dass bestimmte Änderungen an der Apache Tomcat Installation selbst

sind für die ordnungsgemäße Funktion von KNIME Server erforderlich. Diese Modifikationen können Nebenwirkungen haben

zu anderen Webanwendungen, die in der gleichen Installation ausgeführt werden.

Dies sind die Schritte, um KNIME Server in einem benutzerdefinierten Apache Tomcat zu installieren:

ANHANG KNIME installieren Server mit dem KNIME Server Installer.

2. Download und Auszug      [Apache Tomcat 9.0.x](#)      in der gewünschten Version. Nur diese Version von  
Apache Tomcat wird unterstützt.

3. Kopieren Sie die folgenden Dateien aus der von der KNIME erstellten Apache Tomcat Installation

Server Installer in Ihre benutzerdefinierte Apache Tomcat Installation:

- /conf/server.xml
- /conf/userconf.mv.db
- /conf/Catalina/localhost/knime.xml
- /lib/h2-xxx.jar
- /lib/knime-tomcat.jar
- /webapps/knime.war

L 347 vom 20.12.2013, S. 1). Konfigurieren und anpassen der Installation wie in der

[KNIME Server](#)

[Verwaltungshandbuch](#)

Wenn Sie bereits KNIME Server verwenden, können Sie die gleichen Dateien von dieser Installation kopieren plus weitere Dateien, die Sie bei der Anpassung Ihrer Installation erstellt haben (z.B. SSL-Zertifikate oder OIDC Konfigurationsdateien). Stellen Sie sicher, dass Sie Ihre systemd/Windows-Dienste anpassen starten Sie die neue Installation anstelle der alten.



KNIME AG  
Talacker 50  
8001 Zürich, Schweiz  
[www.knime.com](http://www.knime.com)  
[Info@knime.com](mailto:Info@knime.com)