Altair Engineering, Inc. makes no representation or warranties with respect to the contents of this manual or the associated software and especially disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Datawatch Corporation reserves the right to revise this publication and make changes from time to time to its contents without obligation to notify anyone of such revisions or changes.

Altair Knowledge Hub software is offered and is to be used in accordance with a SOFTWARE LICENSE AND MAINTENANCE AGREEMENT. This agreement stipulates that this software be used only in the computer system designated in that agreement. The agreement further stipulates that the customer shall not copy or alter, or permit others to copy or alter, the software or related materials in whole or in part, in any media for any purpose, except to make an archive (back-up) copy or to make a copy as an essential step in the use of the software with the customer’s computer.

Altair Engineering, Inc. hereby grants the buyer the right to reprint this documentation for internal uses only. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, for any other purposes, without the prior written permission of Datawatch Corporation.

Altair Knowledge Hub Installation Guide
Copyright © 2019 by Altair Engineering, Inc.
All rights reserved. Printed in the U.S.A.
Unpublished - Rights reserved under the copyright law of the United States.
Altair Knowledge Hub is a trademark of Altair Engineering, Inc. Other products mentioned herein may be trademarks or registered trademarks of their respective owners in the United States or other countries.

For U.S. Government End Users, the software is a "Commercial Item(s)," as that term is defined at 48 C.F.R. Section 2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. Section 12.212 or 48 C.F.R. Section 227.7202, as applicable. Consistent with 48 C.F.R. Section 12.212 or 48 C.F.R. Sections 227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the Datawatch Software License and Maintenance Agreement.

SALES INFORMATION
US: +1.800.445.3311
International: +1.978.441.2200

Sales Email
US: sales@datawatch.com
Europe: sales_euro@datawatch.com
Asia Pacific: sales_apac@datawatch.com

SUPPORT CONTACT INFORMATION
Customer Portal:
https://support.datawatch.com
Email: support@datawatch.com
US: +1 800.988.4739
International: +1 978.275.8350
Table of Contents

Introduction ........................................................................................................................................... 1

System Requirements .............................................................................................................................. 2
Minimum Requirements ........................................................................................................................... 2
Recommended Specifications ................................................................................................................... 2
Ports ....................................................................................................................................................... 3

Running the Setup Program ................................................................................................................... 4
Post-Installation Steps ............................................................................................................................. 10
Increasing the Heap Size of the Application ........................................................................................... 10
Configuring the Tableau HYPER Component for Export ......................................................................... 10
Log File Locations .................................................................................................................................. 12

Starting Knowledge Hub ......................................................................................................................... 13
Setting Up the HTTPS Protocol ............................................................................................................. 15
Setting Up Single Sign-On (SSO) ........................................................................................................... 20
Setting Up Azure AD Single Sign-On ..................................................................................................... 32
    Registering the Knowledge Hub Application to Azure Active Directory ........................................... 32
    Modifying the Knowledge Hub Config File ....................................................................................... 33
    Using OAuth 2.0 ................................................................................................................................. 35

Configuration File Settings .................................................................................................................... 36
Knowledge Hub Application .................................................................................................................... 36
Knowledge Hub Data Engine ................................................................................................................... 39
Knowledge Hub ML and Spark ................................................................................................................ 43

Upgrading Knowledge Hub .................................................................................................................... 45
Upgrading the Knowledge Hub Application ............................................................................................. 45
Upgrading Java Runtime Environment .................................................................................................... 48

Troubleshooting Knowledge Hub ........................................................................................................... 52
Uninstalling the Application ..................................................................................................................... 52
Introduction

Altair Knowledge Hub (formerly Datawatch Monarch Swarm) allows users across organizations and regions to explore, prepare, and distribute data seamlessly and automatically.

The application allows you to:

- Access data from both structured and semi-structured sources, including PDFs, reports, and all major databases, into orderly tables for subsequent business analytics.
- Open workspaces saved directly from Data Prep Studio in the Swarm library, a local environment, or shared by other users.
- View workspace and table information, including all input data sources, data preparation operations applied, published tables, exports, and schedules, prior to working with it.
- Clean and prepare data for visualization or further analysis.
- Share data from a single-user environment to clustered server environments to deliver the necessary information to vast communities.
- Export data to a number of the most popular visualization and advanced analytics tools.
- Capture data preparations in portable data source definitions, workspaces, and processes to enable easy re-use and sharing across users and environments.
- Schedule process execution for automatic data distribution.
- Enable global collaboration and improve productivity by saving workspaces into the Swarm library for retrieval in Data Prep Studio or sharing workspaces with other Swarm users.
- Socialize data use by sharing and liking objects, subscribing to them and other users, and following users; popular items are recommended to other users based on social interactions and usage patterns.

This installation guide describes the steps necessary to install and access Knowledge Hub.

More information on Knowledge Hub is available via the following links:

- Knowledge Hub Factsheet
- Knowledge Hub Quick Start
- Knowledge Hub Online Help
System Requirements

To run Knowledge Hub successfully, we suggest that your system meet or exceed the minimum requirements specified below. Note that administration rights and permissions are required to install Knowledge Hub. Running the application once installed, however, does not.

Note also that your requirements may vary depending on the volume of data you intend to work with, the number of concurrent users you expect, and several other factors. Contact us to ask about specific requirements for your deployment.

Minimum Requirements

Application Server:

- OS: Windows Server 2016 (recommended), Windows Server 2012 64-bit, or Windows Server 2012 R2 64 bit
- RAM: 32 GB or higher
- Disk Space: 500 GB disk space (depending on the volume of data to be processed)
- CPU: 8 cores or higher
- Java Runtime Environment 8u181 or higher (major version 8, 64-bit)
- For AWS deployments, an m5.2xlarge instance or better

Client Browser (choose one):

- Google Chrome (recommended) – latest version
- Internet Explorer 11
- Mozilla Firefox – latest version
- Microsoft Edge – latest version

Recommended Specifications

Knowledge Hub is a highly scalable and flexible application. We strongly advise contacting your Altair account manager to obtain recommended specifications for the deployment you wish to implement.
Ports

The following ports must be opened for Knowledge Hub to function correctly:

- If you are using HTTP, open port **8080**
- If you are using HTTPS, open port **8443**

When setting up Knowledge Hub on the HTTPS protocol you must map localhost to the host name. However, if you choose not to, the following ports should be opened:

- Port **9091** must be opened to allow the socialization and machine-learning components of the application to work correctly. When installing the application on AWS, port 9091, as well as all other ports (i.e., 8080 or 8443, 4040) must be added to your AWS instance security group and the port must be open in the server instance.
- Port **4040** must be opened to allow Spark to work correctly.
- Port **8081** must be opened for proper communication between the Altair Knowledge Hub service and the Altair Knowledge Hub Data Engine service.

The protocol type for these ports is TCP.

Note that if mapping localhost to the host name is not done, Knowledge Hub may be installed without errors but usual activities, such as specifying settings or importing a data source, may yield multiple unexpected error messages.
Running the Setup Program

The steps below describe how to install Knowledge Hub in a Windows environment.

Steps:

1. Manually download the [Windows 64-bit version of Java SE Runtime Environment 8](https://www.oracle.com/java/technologies/javase-jre8.html) and install it.
   
   Note that while the minimum version required by Knowledge Hub is update 181, newer updates are also supported by the installer (e.g., 211 for minor versions).

2. Extract the contents of the zip file you obtained from the download link provided by your product specialist. This file should contain:
   
   - The application installer
   - A prerequisites folder

   The prerequisites folder should include installers for:
   
   - Altair JDBC Drivers
   - PostgreSQL 11
   - A libs folder containing CDATA JDBC drivers to enable the creation of connections

   If any of these items is not in the file you downloaded, contact your Altair product specialist immediately.

   Note that version 3.8.3 of the Snowflake JDBC driver may cause issues when using Snowflake as a custom connection. To address this issue, we recommend using the 3.6.9 version of the Snowflake JDBC driver instead.

3. Your license will be provided to you by Altair. Rename this file to `license.lic` and copy it to the folder containing the Knowledge Hub prerequisites folder and installer.

   You may also opt to implement HyperWorks Units-based licensing. To do so, create a new environment variable with the name ALTAIR_LICENSE_PATH and set its value to "<license server port>@<license server host> (e.g., 6200@100.0.0.100).

   Note that only one licensing system can be used at any one time.

4. Double click on the application installer (`DatawatchMonarchSwarm.exe`) to run it.

5. In the next screen, read the license agreement. If you agree with its terms, select I accept the agreement and then click Next.
6. The Configuration screen allows you to specify whether the HTTP or HTTPS protocol should be used. The default port for each protocol is also indicated. Choose the protocol you would like to apply. Unless you want to change them, accept the default port values indicated and then click Next.

If you wish to use a different set of ports, enter them into the appropriate fields provided and then click Next. Note that the ports you specify must not be used by other applications.

7. The Database Configuration screen allows you to specify a database name, a database user name, and a database user password. The default for all three items is newserver. If you wish to create a new database, user, and password, enter the required names and passwords into the corresponding fields.
The *Database Configuration* screen requires you to specify a PostgreSQL administrator login and password to use for new PostgreSQL installations.

Take note of the administrator login and password you supply as these details will be required when you upgrade the Knowledge Hub application. If preexisting installation of PostgreSQL is available, the administrator login and password should be provided here.

Enter the necessary details and then click **Next**.

8. The *Select Destination Location* screen allows you to specify a folder in which the application will be installed. If you do not wish to change the default location indicated, click **Next**. Otherwise, specify a new location and then click **Next**.
9. The Select Components screen allows you to specify which Knowledge Hub components to install.

Deselect the component(s) you do not wish to install and then click Next.

10. The Select Start Menu Folder screen allows you to specify a Start Menu folder into which the program's shortcuts will be placed. Change it to a different folder if you wish and then click Next or simply click Next to accept the default value indicated.
11. The *Ready to Install* screen provides a summary of your installation folders. Click **Install** to begin installing Knowledge Hub.

![Ready to Install screen](image1.png)

The installation begins.

![Installing screen](image2.png)

12. When installation has completed, the following message displays. In some cases, you may be asked to restart your computer. Click **Finish** to do so.

![Installing screen](image2.png)
When the application is correctly installed, the following services become available:

- Datawatch Monarch Swarm
- Datawatch Monarch Swarm Data Engine
- Datawatch Monarch Swarm ML
- Datawatch Monarch Swarm Spark

Note that these services refer to the Knowledge Hub, Knowledge Hub Data Engine, Knowledge Hub ML, and Knowledge Hub Spark services, respectively.

Note also that while Postgres and Cassandra are automatically installed when Knowledge Hub is installed, these applications are reserved for Knowledge Hub use only. Thus, addition of custom tables/data is not supported.

Potential Issues with Older Versions of Microsoft Visual C++ Redistributable

When an older version of Microsoft Visual C++ (VC++) Redistributable (e.g., 2012 and older) is installed in a server in which Knowledge Hub is also to be installed, the following error messages may display:

This issue is due to an incompatibility between the older redistributable package and the VC++ version installed with the application. To address this issue, simply delete the older version of VC++ and reinstall the application.
Post-Installation Steps

INCREASING THE HEAP SIZE OF THE APPLICATION

The following procedure represents an important and mandatory post-installation configuration step to increase the heap size of the application and maximize performance. These instructions should be performed by an experienced system administrator.

Note that heap size settings should also be reapplied after an upgrade is performed.

Steps:

1. Calculate the maximum heap size to allocate to the application. While requirements may vary according to the environment in which the application is being deployed, a heap size of 75% of the physical memory of the server is recommended.

   For example, if the physical memory of the server is 64 GB, the maximum heap size should be **48 GB**.

2. Modify `monarchswarm.xml` and `monarchswarm-dp.xml` as follows, replacing the numerical value of the parameter “-Xmx16g” with the value calculated in Step 1:

   `-Xms4g -Xmx16g`

   For example:

   ```xml
   <arguments>-jar "core-api.jar" -Xms4g -Xmx16g --spring.profiles.active=prod</arguments>
   ``

   where:

   “-Xms4g” and “-Xmx16g” indicate the minimum and maximum heap sizes, respectively.

CONFIGURING THE TABLEAU HYPER COMPONENT FOR EXPORT

Previous versions of Knowledge Hub supported both the Tableau SDK (TDE) and Tableau Hyper (HYPER) components for exporting data. In the current version of Knowledge Hub, only exports to the Tableau Hyper component are supported. This component is installed by default in Knowledge Hub v2.3. If the application is installed via upgrade from an earlier version with a customized Tableau SDK, your Environment Variables must be modified to switch support for Tableau Hyper.

Note that older workspaces containing data sources from Tableau SDK are not affected by this change.
Steps:

1. In Windows, launch the System Properties dialog (typically by right-clicking on **Computer > Properties > Advanced System Settings**) and then click **Environment Variables**.

   The Environment Variables dialog displays.

   ![Environment Variables dialog]

2. Edit the **Path** property in the **System variables** section of the **Environment Variables** dialog.

3. From the list of system variables, delete `C:\Program Files\Datawatch Monarch Swarm\tableau_sdk\bin` and leave `C:\Program Files\Datawatch Monarch Swarm\tableau_hyper\bin`.

4. Save changes and restart Windows.
Log File Locations

Knowledge Hub log files may be found in the following locations:

<table>
<thead>
<tr>
<th>LOG FILE</th>
<th>LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cassandra</td>
<td>&lt;Knowledge Hub folder&gt;\cassandra\logs</td>
</tr>
<tr>
<td>Knowledge Hub service (core-api.log)</td>
<td>%TEMP%\MonarchSwarm\Logs\</td>
</tr>
<tr>
<td>Knowledge Hub Data Engine service (data-engine-api.log)</td>
<td>e.g., C:\Windows\Temp\MonarchSwarm\Logs</td>
</tr>
<tr>
<td>Knowledge Hub ML and Spark services (ml-app.log)</td>
<td></td>
</tr>
<tr>
<td>Installation log file</td>
<td>The installation log file is saved in the user’s Temp folder (i.e., the person who ran the installer) e.g., C:\Users&lt;User's name&gt;\AppData\Local\Temp Installation log files are named as follows: Setup Log YYYY-MM-DD #NNN.next where: YYYY – current year MM -current month DD – current day NNN – a unique number provided to each installation starting from 1</td>
</tr>
</tbody>
</table>
Starting Knowledge Hub

If you opt not to restart your computer immediately after installing the application, the Knowledge Hub login page displays as soon as you click **Finish** on the final screen of the installer.

If you opted to restart your computer after installation or if you are accessing Knowledge Hub from a different machine, launch your browser, enter the IP address of the machine you are connecting to into the address bar (i.e., “10.0.0.101”), and then append the application port specified during installation (in this case “:8080”) to it (e.g., “10.0.0.101:8080”). Press **Enter** on your keyboard afterward.
The Knowledge Hub login page displays.

If you selected the HTTPS protocol, your address bar should display as follows:

You may need to modify the firewall settings of your server if an error page, such as that shown below, displays when trying to access the Knowledge Hub login page. Contact your network administrator if this problem persists.
Login using the credentials provided to you by your product specialist and then click **Sign in**. The Knowledge Hub dashboard displays.

### Setting Up the HTTPS Protocol

An SSL certificate is required to run Knowledge Hub installations using the HTTPS protocol. **Note that this procedure is best performed by a knowledgeable system administrator.**

In the steps outlined below, the keystore file `key-store.jks` and the security certificate `key-store.cer` are generated. You can generate the keystore and certificate using any name you wish. Note also that generating a self-signed certificate is not the ideal way of enabling the HTTPS protocol. A proper certificate from a Certificate Authority or signing a new certificate from the corporate root certificate is recommended.

**Steps:**

1. Generate a security certificate.

   1.1 Go to the bin directory of JRE (i.e., the path where Java is installed, usually `C:\Program Files\Java\jre1.8.x xx\bin`), paste the following script and run it. In this example, replace "localhost" with your domain name in CN=**localhost**. Replace "localhost" with the machine name in san=dns:**localhost**. Alternatively, change the value in ip to the IP of the machine.
keytool -genkey -alias server -keyalg RSA -keysize 4096 -keystore <your_keystore_filename>.jks -validity 3652 -dname "CN=<domain_name>, O=<your_company_name>, L=<your_location>, ST=<your_state>, C=<your_country>" -ext san=dns:<your_machine_name.domain_name>,ip:<Knowledge_Hub_server_IP> && keytool -certreq -alias server -file key-store.csr -keystore <your_keystore_filename>.jks

where:

- <your_keystore_filename> - the name of the keystore file used to generate the certificate, e.g., key-store
- <domain_name> - the domain to which the client machine is joined, e.g., swarm.altair.com
- <your_company_name> - the name of the company to which you belong, e.g., Altair Engineering
- <your_location> - the company location, e.g., Bedford
- <your_state> - the state in which your company is located, e.g., MA
- <your_country> - the country in which your company is located, e.g., US
- <your_machine_name.domain_name> - the name of the client machine, e.g., machinename1.swarm.altair.com
- <Knowledge_Hub_server_IP> - the IP address of the server in which Knowledge Hub is installed; in the present example, the value 127.0.0.1 is used, and this value must be added to the host file (Step 2.5)

Thus, a complete script may look like:


To obtain more information on keytool, run keytool -help or go to https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html.

1.2 You will be asked to set a **keystore password** and a **key password**. These passwords should be the same. You will be asked to repeat your keystore and key passwords at a later time so take note of this information.

1.3 Run the following command to export the certificate:

keytool -export -alias server -keystore key-store.jks -rfc -file key-store.cer

A security certificate is generated in C:\Program Files\Java\jre1.8.0_XXX\bin\.

1.4 Copy the keystore file (*.jks) to the **certificate** folder of the Knowledge Hub directory (C:\Program Files\Datawatch Monarch Swarm\certificate).
2. Configure Knowledge Hub.

2.1 Edit the Knowledge Hub configuration file.

2.1.1 Open application-prod.yml. This file is normally located in C:\Program Files\Datawatch Monarch Swarm.

2.1.2 Set the following parameters:

```yaml
server:
  port: 8443
  ssl:
    key-store: key-store.jks # Path to generated keystore
    key-store-password: Pa$$word # Keystore password
    key-password: Pa$$word # Key password
    enabled: true

application:
  data-engine-api:
    url: https://<machine_name>:8081
```

Parameter settings:
- server.port – The port to use for the HTTPS protocol (8443)
- server.ssl.key-store – The path to the generated keystore
- server.ssl.key-store-password – The keystore password
- server.ssl.key-password – The key password
- server.ssl.enabled – Set this item to true to enable SSL

2.2 Edit the Knowledge Hub ML configuration file.

2.2.1 Open social-application.yml. This file is normally located in C:\Program Files\Datawatch Monarch Swarm\ml.

2.2.2 Set the following parameters:

```yaml
server:
  ssl:
    enabled: true
    keystore.password: Pa$$word # Keystore password
    keystore.path: ..\\certificate\\key-store.jks # Path to the generated keystore
```
Parameter settings:

- server.ssl.enabled – Set this item to true to enable SSL
- server.ssl.key-store.password – The keystore password
- server.ssl.key-store.path – The path to the generated keystore

2.3 Edit the Knowledge Hub Data Engine configuration file.

2.3.1 Open application-prod.yml in the dp folder (C:\Program Files\Datawatch Monarch Swarm\dp).

2.3.2 Set the following parameters:

```yaml
server:
  ssl:
    enabled: true
    key-store: ..\\certificate\\key-store.jks
    key-store-password: Pa$$word # Keystore password
    key-password: Pa$$word # Key password
```

```yaml
application:
  core-api:
    url: https://machine_name:8443
```

Parameter settings:

- server.ssl.enabled – Set this item to true to enable SSL
- server.ssl.key-store – The path to the generated keystore
- server.ssl.key-store-password – The keystore password
- server.ssl.key-password – The key password

2.4 Import the generated certificate to the Java keystore.

2.4.1 Open Command Prompt with administrator rights, paste the following script, and then run it.

```bash
keytool -import -noprompt -v -trustcacerts -alias datawatch_monarch_swarm_certificate -file "C:\Program Files\Datawatch Monarch Swarm\certificate\key-store.cer" -keystore "C:\Program Files\Java\jre1.8.x_xx\lib\security\cacerts" -storepass changeit
```

Note that the path to the Java keystore is usually C:\Program Files\Java\jre1.8.x_xx\lib\security\cacerts and the path to the certificate file is usually C:\Program Files\Datawatch Monarch Swarm\certificate\key-store.cer.

2.4.2 If the certificate is renewed, delete the previously imported certificate from the Java keystore using the script below and then import the new certificate as described above.
keytool -delete -alias datawatch_monarch_swarm_certificate -keystore "C:\Program Files\Java\jre1.8.x_xx\lib\security\cacerts" -storepass changeit

2.5 Add to the hosts file (usually located in C:\Windows\System32\drivers\etc) the mapping of the localhost to the host name.

e.g., 127.0.0.1 host name

2.6 Restart the Knowledge Hub, Knowledge Hub Data Engine, Knowledge Hub Spark, and Knowledge Hub ML services.

3. Configure clients.

3.1 Add the generated certificate to Trusted Root on all machines on which Knowledge Hub will be used.

3.1.1 Run Certificate Manager.

3.1.2 Import the files key-store.jks and key-store.cer to Trusted Root Certification Authorities.

3.2 Launch Knowledge Hub in your browser using the following form:
https://<ipaddress>:8443.
Setting Up Single Sign-On (SSO)

Single Sign-On (SSO) can be set up to log into Knowledge Hub to implement organizational security and allow a larger number of users to access the application without needing to create a user profile for each one.

**Note that the following steps are best performed by a knowledgeable system administrator.**

The steps outlined in this section assume that:

- There exists a domain controller server in which Lightweight Directory Access Protocol (LDAP) has been enabled and configured.
- There exists a server in which Knowledge Hub has been installed. This server must be part of the domain.
- There exists a user with a client machine who is a member of the domain for which Knowledge Hub is being configured.

Note that the scripts to be run and config file are extremely sensitive to formatting and case. Exercise caution when running the necessary scripts and making additions to the Knowledge Hub config file.

Steps:

**A. On the server with Active Directory Domain Controller:**

1. In the machine with the AD domain controller, create an AD user (e.g., **tomcat**) with a password (e.g., **Password#**).

2. Run the following scripts in Powershell in Administrator mode:

   ```
   * setspn -A HTTP/<full computer name of Knowledge Hub server> tomcat
   * ktpass /out c:\tomcat.keytab /mapuser tomcat@<DOMAIN_NAME> /princ HTTP/<full computer name of Knowledge Hub server>@<domain name> /pass Password# /ptype KRB5_NT_PRINCIPAL /crypto All
   ```

   For example, assuming that **WIN-SWARMSERVER.altair.com** is the complete computer name of the Knowledge Hub server and the domain name is **altair.com**, the following scripts should be run:

   ```
   * setspn -A HTTP/WIN-SWARMSERVER.altair.com tomcat
   * ktpass /out c:\tomcat.keytab /mapuser tomcat@ALTAIR.COM /princ HTTP/WIN-SWARMSERVER.altair.com@ALTAIR.COM /pass Password# /ptype KRB5_NT_PRINCIPAL /crypto All
   ```
## B. On the Knowledge Hub server:

1. Stop all of the related Knowledge Hub services.
2. Copy the file `tomcat.keytab` generated in Step A2 and place it in `C:/Users/tomcat`.
3. Open the file `application-prod.yml`, which is typically located in `C:/Program Files/Datawatch Monarch Swarm`, and then add the following properties.

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>application.security.authentication.provider</td>
<td>Use <code>ldap</code> for LDAP/SSO; may also use <code>basic</code> for basic and <code>oauth2</code> for OAuth2 authentication</td>
</tr>
<tr>
<td>application.security.authentication.users-provisioned</td>
<td>Enables (true) or disables (false) explicit user provisioning. If explicit provisioning is disabled, the system creates Knowledge Hub users automatically. When set to true, users must be created manually</td>
</tr>
<tr>
<td>application.security.authentication.default-password</td>
<td>The default password for new users created through LDAP and added multiple users</td>
</tr>
<tr>
<td>application.security.authentication.sso.enabled</td>
<td>true to enable SSO; false if using LDAP authentication</td>
</tr>
<tr>
<td>application.security.authentication.sso.service-principal</td>
<td>Full computer name of the Knowledge Hub server in the form HTTP/&lt;COMPUTER NAME&gt;.&lt;domain&gt;/@&lt;DOMAIN&gt; (e.g., HTTP/WIN-SWARMSERVER. <a href="mailto:altair.com@ALTAIR.COM">altair.com@ALTAIR.COM</a>)</td>
</tr>
<tr>
<td>application.security.authentication.sso.key-tab-location</td>
<td>Path to keytab file (e.g., C:/Users/tomcatone.keytab)</td>
</tr>
<tr>
<td>application.security.authentication.sso.request-regex</td>
<td>^/api/.+/ldap_sso - Setting for SSO</td>
</tr>
<tr>
<td>application.security.authentication.ldap.query.attribute-mapping:</td>
<td>Attributes used to add users by LDAP query</td>
</tr>
<tr>
<td>login: userPrincipalName</td>
<td></td>
</tr>
<tr>
<td>first-name: givenname</td>
<td></td>
</tr>
<tr>
<td>last-name: sn</td>
<td></td>
</tr>
<tr>
<td>common-name: cn</td>
<td></td>
</tr>
<tr>
<td>email: mail</td>
<td></td>
</tr>
<tr>
<td>phone-number: telephonenumber</td>
<td></td>
</tr>
<tr>
<td>PROPERTYS</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>groups: memberOf</td>
<td></td>
</tr>
<tr>
<td>application.security.authentication.sso.query</td>
<td></td>
</tr>
<tr>
<td>custom-attributes:</td>
<td></td>
</tr>
<tr>
<td>- displayName</td>
<td></td>
</tr>
<tr>
<td>- distinguishedName</td>
<td></td>
</tr>
<tr>
<td>- name</td>
<td></td>
</tr>
<tr>
<td>- objectCategory</td>
<td></td>
</tr>
<tr>
<td>- objectClass</td>
<td></td>
</tr>
<tr>
<td>- primaryGroupID</td>
<td></td>
</tr>
<tr>
<td>- sAMAccountName</td>
<td></td>
</tr>
<tr>
<td>- sAMAccountType</td>
<td></td>
</tr>
<tr>
<td>- servicePrincipalName</td>
<td></td>
</tr>
<tr>
<td>application.security.authentication.ldap.active-directory</td>
<td>true when AD is used</td>
</tr>
<tr>
<td>application.security.authentication.ldap.domain</td>
<td>Domain name</td>
</tr>
<tr>
<td>application.security.authentication.ldap.domain-users</td>
<td>Allow LDAP authentication for any of two forests in one domain. The default value for this setting is false. To authenticate users from just one domain via LDAP, set this property to true and then set the correct domain in the property application.security.authentication.ldap.domain</td>
</tr>
<tr>
<td>application.security.authentication.ldap.server</td>
<td>Full computer name of LDAP server</td>
</tr>
<tr>
<td>application.security.authentication.ldap.manage-dn</td>
<td>User name and password to use to connect to LDAP server when sso.enabled = false and ldap-active-directory = false</td>
</tr>
<tr>
<td>application.security.authentication.ldap.manage-password</td>
<td>If sso.enabled = true and ldap-active-directory = true, these properties may be omitted from the config file. These credentials are also used to add multiple users to Knowledge Hub using LDAP query</td>
</tr>
<tr>
<td>application.security.authentication.ldap.search-base</td>
<td>Domain name components (e.g., DC=altair,DC=com if domain is altair.com)</td>
</tr>
<tr>
<td>PROPERTY</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>application.security.authentication.ldap.search-filter</td>
<td>Filter used to search for LDAP users</td>
</tr>
<tr>
<td>application.security.authentication.ldap.user-roles</td>
<td>User role(s) for SSO users</td>
</tr>
<tr>
<td>application.security.authentication.ldap.admin-users</td>
<td>List of users automatically created with the Super Administrator role in Knowledge Hub (if users-provisioned: false). When this list is provided, there is no need to login as an administrator and create the first LDAP user.</td>
</tr>
<tr>
<td>application.security.authentication.ldap.role-mapping</td>
<td>true to enable role-mapping in Knowledge Hub; false to disable</td>
</tr>
<tr>
<td>application.security.authentication.ldap.group-mapping</td>
<td>true to enable group-mapping in Knowledge Hub; false to disable</td>
</tr>
<tr>
<td>application.security.authentication.ldap.roles-map</td>
<td>Mapping of Knowledge Hub roles to LDAP groups</td>
</tr>
</tbody>
</table>

For example, using the domain and Knowledge Hub server names specified above and assuming that **WIN-LDAPSERVER.altair.com** is the computer name of the domain controller, the following strings should be added:

```yaml
authentication:
  provider: ldap
  users-provisioned: true
  default-password: password
ldap:
  sso:
    enabled: true
    service-principal: HTTP/WIN-SWARMSERVER.altair.com/@ALTAIR.COM
  key-tab-location: C:/Users/tomcat/tomcat.keytab
  request-regex: ^/api/.+/ldap_sso
query:
  attribute-mapping:
    login: userPrincipalName
    first-name: givenname
    last-name: sn
    common-name: cn
    email: mail
    phone-number: telephonenumber
    groups: memberOf
  custom-attributes:
    - displayName
```
- distinguishedName
- name
- objectCategory
- objectClass
- primaryGroupID
- sAMAccountName
- SAMAccountType
- servicePrincipalName

active-directory: true
domain-users: false
domain: altair.com
server: ldap://WIN-LDAPSERVER.altair.com/
manage-dn: swarmadmin@altair.com
manage-password: Passw0rd#
search-base: DC=altair,DC=com
search-filter: "(| (userPrincipalName={0}) (sAMAccountName={0}))"
user-roles:
- 3
admin-users:
- mbarnes@altair.com
- tjones@altair.com

Note that the item "user-roles: - 3" (this item MUST span two lines) indicates that the created user is assigned the role ANALYST. You must indicate a role for users in the config file as you cannot do so in the application. You can change the user role by changing the number indicated to any of the following:

- 1 – ADMIN
- 2 – CONSUMER
- 3 – ANALYST
- 4 – CURATOR
- 5 – DESKTOP
- 6 – ADVANCED
- 7 – DATA SCIENTIST
- 8 – SUPER ADMINISTRATOR

You can also assign the user multiple roles as in the following:

    user-roles:
    - 3
    - 4

This code assigns the user the roles ANALYST and CURATOR. Note that, in this case, this snippet must span 3 lines. More information on user roles may be found here.
4. Restart the application using the AD user **tomcat**.

**C. On the client machine:**

1. Log in using the AD username and password of a user who is a member of the domain.
2. If necessary, configure your browser to accept and trust domain sites (i.e., add to local intranet and trusted sites; Google Chrome does not require this).
3. Launch the Knowledge Hub login page using the form **https://<full name of Knowledge Hub server>:8443** (in the present example, we would enter **https://win-swarmserver.altair.com:8443** into the address bar of our browser) and then click the **SSO Sign In** button.

A login dialog displays.
4. Enter the username and password of your user and then click the **Log in** button.

Once logged in, the Knowledge Hub dashboard displays:

A look at the user’s profile should display as follows:
Future sessions using the same client machine will no longer require the user to log in with his/her credentials.


Knowledge Hub supports LDAP authentication. This section describes how LDAP authentication can be implemented in Knowledge Hub.

**Note that the following steps are best performed by a knowledgeable system administrator.**

Steps:

1. Install Knowledge Hub as usual.
2. Stop all Knowledge Hub services.
3. Open the file **application-prod.yml**, which is typically located in C:/Program Files/Datawatch Monarch Swarm, and then add the following properties:
<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>application.security.authentication.provider</td>
<td>Use <code>ldap</code> for LDAP/SSO; may also use <code>basic</code> for basic and <code>oauth</code> for OAuth2 authentication</td>
</tr>
<tr>
<td>application.security.authentication.users-provisioned</td>
<td>Enables (true) or disables (false) explicit user provisioning. If explicit provisioning is disabled, the system creates Knowledge Hub users automatically. When set to true, users must be created manually</td>
</tr>
<tr>
<td>application.security.authentication.default-password</td>
<td>The default password for new users created through LDAP and added multiple users</td>
</tr>
<tr>
<td>application.security.authentication.ldap.query</td>
<td>Attributes used to add users by LDAP query</td>
</tr>
<tr>
<td>attribute-mapping:</td>
<td></td>
</tr>
<tr>
<td>login: userPrincipalName</td>
<td></td>
</tr>
<tr>
<td>first-name: givenname</td>
<td></td>
</tr>
<tr>
<td>last-name: sn</td>
<td></td>
</tr>
<tr>
<td>common-name: cn</td>
<td></td>
</tr>
<tr>
<td>email: mail</td>
<td></td>
</tr>
<tr>
<td>phone-number: telephonenumber</td>
<td></td>
</tr>
<tr>
<td>groups: memberOf</td>
<td></td>
</tr>
<tr>
<td>application.security.authentication.sso.query</td>
<td></td>
</tr>
<tr>
<td>custom-attributes:</td>
<td></td>
</tr>
<tr>
<td>- displayName</td>
<td></td>
</tr>
<tr>
<td>- distinguishedName</td>
<td></td>
</tr>
<tr>
<td>- name</td>
<td></td>
</tr>
<tr>
<td>- objectCategory</td>
<td></td>
</tr>
<tr>
<td>- objectClass</td>
<td></td>
</tr>
<tr>
<td>- primaryGroupID</td>
<td></td>
</tr>
<tr>
<td>- sAMAccountName</td>
<td></td>
</tr>
<tr>
<td>- sAMAccountType</td>
<td></td>
</tr>
<tr>
<td>- servicePrincipalName</td>
<td></td>
</tr>
<tr>
<td>application.security.authentication.ldap.active-directory</td>
<td><strong>true</strong> when AD is used</td>
</tr>
<tr>
<td>PROPERTY</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>application.security.authentication.ldap.domain</td>
<td>Domain name</td>
</tr>
<tr>
<td>application.security.authentication.ldap.domain-users</td>
<td>Allow LDAP authentication for any of two forests in one domain. The default value for this setting is <strong>false</strong>. To authenticate users from just one domain via LDAP, set this property to <strong>true</strong> and then set the correct domain in the property application.security.authentication.ldap.domain</td>
</tr>
<tr>
<td>application.security.authentication.ldap.server</td>
<td>Full computer name of LDAP server</td>
</tr>
<tr>
<td>application.security.authentication.ldap.manage-dn</td>
<td>User name and password to use to connect to LDAP server when sso.enabled = <strong>false</strong> and ldap-active-directory = <strong>false</strong></td>
</tr>
<tr>
<td>application.security.authentication.ldap.manage-password</td>
<td>If sso.enabled = <strong>true</strong> and ldap-active-directory = <strong>true</strong>, these properties may be omitted from the config file. These credentials are also used to add multiple users to Knowledge Hub using LDAP query</td>
</tr>
<tr>
<td>application.security.authentication.ldap.search-base</td>
<td>Domain name components (e.g., DC=altair,DC=com if domain is altair.com)</td>
</tr>
<tr>
<td>application.security.authentication.ldap.search-filter</td>
<td>Filter used to search for LDAP users</td>
</tr>
<tr>
<td>application.security.authentication.ldap.user-roles</td>
<td>User role(s) for SSO users</td>
</tr>
<tr>
<td>application.security.authentication.ldap.admin-users</td>
<td>List of users automatically created with the Super Administrator role in Knowledge Hub (if users-provisioned: <strong>false</strong>). When this list is provided, there is no need to login as an administrator and create the first LDAP user.</td>
</tr>
<tr>
<td>application.security.authentication.ldap.role-mapping</td>
<td>true to enable role-mapping in Knowledge Hub; <strong>false</strong> to disable</td>
</tr>
<tr>
<td>application.security.authentication.ldap.group-mapping</td>
<td>true to enable group-mapping in Knowledge Hub; <strong>false</strong> to disable</td>
</tr>
<tr>
<td>application.security.authentication.ldap.roles-map</td>
<td>Mapping of Knowledge Hub roles to LDAP groups</td>
</tr>
</tbody>
</table>
Note the following behaviors:

- If the property `users-provisioned` is set to `true`, and the user is not included in the `admin-users` list, an error (i.e., "User %user_login% does not exist") is returned when the user logs into the application via SSO. In this case, the user must be manually added through the User Management page (via LDAP) of Knowledge Hub.

- If the property `users-provisioned` is set to `false`, and the user exists in Active Directory, a new user is created upon login to Knowledge Hub via SSO. This user's profile will include a login, last name, and first name, and s/he will have the role(s) specified in `user-roles`.

- If the user exists in Active Directory, and the new user is included in the `admin-users` list, the user can log into Knowledge Hub via SSO and this user will have the role Super Administrator regardless if the property `users-provisioned` is set to either `true` or `false`.

For example, using the domain and Knowledge Hub server names specified above and assuming that `WIN-LDAPSERVER.altair.com` is the computer name of the domain controller, the following strings should be added:

```
authentication:
  provider: ldap
  users-provisioned: true
  default-password: password
  ldap:
    query:
      attribute-mapping:
        login: userPrincipalName
        first-name: givenname
        last-name: sn
        common-name: cn
        email: mail
        phone-numer: telephonenumber
        groups: memberOf
      custom-attributes:
        - displayName
        - distinguishedName
        - name
        - objectCategory
        - objectClass
        - primaryGroupID
        - sAMAccountName
        - SAMAccountType
        - servicePrincipalName
  active-directory: true
  domain-users: false
```
domain: altair.com
server: ldap://WIN-LDAPSERVER.altair.com/
manage-dn: swarmadmin@altair.com
manage-password: Passw0rd#
search-base: DC=altair,DC=com
search-filter: "(| (userPrincipalName={0}) (sAMAccountName={0}))"
user-roles:
- 3
admin-users:
- mbarnes@altair.com
- tjones@altair.com

In the present example, two users (i.e., new_user1 and new_user2) are assigned the Super Administrator roles.

The following example shows settings if you wish to implement role/group mapping:

application:
server:
  internet-address: https://knowledgehub.altair.com:8443
security:
  authentication:
    provider: basic
    xauth:
      secret: Dkdi4kdk3kdalp184sAneCxpYJba038134
    ldap:
      group-mapping: true
      role-mapping: true
      roles-map:
        3:
          - "TestGroup1"
          - "123BLK"
          - "991BLK"
        5:
          - "TestGroup2"
          - "123Blk"

4. Restart all Knowledge Hub services.
Setting Up Azure AD Single Sign-On

SSO to Knowledge Hub may be accomplished through OAuth 2.0, which can be configured using several identity providers, including Azure Active Directory. As in previous cases, user provisioning can be created either automatically or explicitly.

**Note that this procedure is best performed by a knowledgeable system administrator.**

The steps outlined in this section assume that:

- A user from Azure AD is mapped to a Knowledge Hub user by login.
- A user with such a login exists in Knowledge Hub.

The following steps summarize how to set up OAuth 2.0 and use with Azure AD.

**REGISTERING THE KNOWLEDGE HUB APPLICATION TO AZURE ACTIVE DIRECTORY**

Steps:

1. Sign in to the Azure portal.
2. In the left-hand navigation pane, select the Azure Active Directory service, and then select App registrations (Preview) > New registration.
3. When the Register an application page appears, enter your application's registration information:
   - **Name** - Enter a meaningful application name that will be displayed to users of the app.
   - **Supported account types** - Select which account you would like your application to support.
     - **Accounts in this organizational directory only** – Select this option if you are building a line-of-business (LOB) application. This option is not available if you are not registering the application in a directory. This option maps to Azure AD only single-tenant.
     - **Accounts in any organizational directory** - Select this option if you would like to target all business and educational customers. This option maps to an Azure AD only multi-tenant.
     - **Accounts in any organizational directory and personal Microsoft accounts** - Select this option to target the widest set of customers. This option maps to Azure AD multi-tenant and personal Microsoft accounts.
   - **Redirect URI** – `<kh-host>/login/oauth2/code/knowledgehub`
4. When finished, select **Register**.

5. Copy the **Application ID** from the app's **Overview** page. This ID is the **client_id**, which you will need to modify the config file.

6. Select the **Certificates & Secrets** section from the app's **Overview** page.

7. Select New client secret.

8. Add a description for your client secret, select a duration, and then click **Add**.

9. After saving the configuration changes, the right-most column will contain the **client_secret** value, which you will need to update the Knowledge Hub config file.

More information on registering an app on Azure AD can be found in [https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-register-app](https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-register-app).

MODIFYING THE KNOWLEDGE HUB CONFIG FILE

Open the file **application-prod.yml**, which is typically located in C:/Program Files/Datawatch Monarch Swarm, and then add the following properties.

```yaml
application:
  server:
    internet-address: https://<server address>
  security:
    authentication:
      provider: oauth2
      oauth2:
        client:
          clientId: <OAUTH2_CLIENT_ID>
          clientSecret: "<OAUTH2_CLIENT_SECRET>"
          accessTokenUri: https://login.microsoftonline.com/common/oauth2/token
          userAuthorizationUri: https://login.microsoftonline.com/common/oauth2/authorize
          jwtkuri: https://login.microsoftonline.com/common/discovery/keys
          scope: openid,https://graph.microsoft.com/user.read
          resource:
            userInfoUri: https://graph.microsoft.com/v1.0/me
      user-roles:
        - 2
      admin-users:
        attribute-mapping:
          login: upn
          first-name: given_name
          last-name: family_name
          email: upn
          phone-number: telephonenumber
```
where:

- **application.server.internet-address** – is URL of the Knowledge Hub server
- **application.security.authentication.oauth2.client.clientId** – is the client ID
- **application.security.authentication.oauth2.client.clientSecret** – is the client secret
- **application.security.authentication.oauth2.user-roles** – default role of users added to Knowledge Hub
- **application.security.authentication.oauth2.admin-users** – users with the Super Administrator role in Knowledge Hub
- **application.security.authentication.oauth2.attribute-mapping** – settings used to add users by Azure AD query

For example:

```yaml
application:
  server:
    internet-address: https://win-swarmsserver.altair.com:8080
  security:
    authentication:
      provider: oauth2
      users-provisioned: false
      default-password: defaultPassword
    oauth2:
      client:
        clientId: 123456a-78bc-9d10-e11f-1213gade9c3
        clientSecret: =Q7abcS[d=]e2F3GhiJkJlMNoP4.q2r8S
      accessTokenUri: https://login.microsoftonline.com/common/oauth2/token
      userAuthorizationUri: https://login.microsoftonline.com/common/oauth2/authorize
      jwkUri: https://login.microsoftonline.com/common/discovery/keys
      scope: openid,https://graph.microsoft.com/user.read
      resource:
        userInfoUri: https://graph.microsoft.com/v1.0/me
      user-roles:
        - 3
    admin-users: admin@altair.com
  attribute-mapping:
    login: upn
    first-name: given_name
    last-name: family_name
    email: upn
    phone-number: telephonenum
```
USING OAUTH 2.0

- GET http://<kh-host>/api/latest/config

RESPONSE

```json
{
    "application.data-engine.store.design-mode-limit": "10000",
    "application.security.authentication.ldap.sso.enabled": "false",
    "application.io.connection.disabled": "",
    "application.security.authentication.provider": "oauth2",
    "application.security.authentication.oauth2.authorization-url": "text"
}
```

- If application.security.authentication.provider equals oauth2, then open <kh-host>/<application.security.authentication.oauth2.authorization-url>
Configuration File Settings

Knowledge Hub Application

Knowledge Hub’s configuration file, named `application-prod.yml`, is typically created and stored in C:\Program Files\Datawatch Monarch Swarm. A snippet of the contents of this file is shown below.

```yaml
spring:
  data:
    cassandra:
      enabled: true
  datasource:
    url: jdbc:postgresql://localhost:1000/newserver
    username: newserver
    password: newserver
  http:
    multipart:
      maxFileSize: 2000MB
      maxRequestSize: 2000MB
  server:
    port: 8080
    ssl:
      enabled: false
      key-store: certificate\localhost.jks
      key-store-password: password
      key-password: password
    tomcat:
      accesslog:
        enabled: true
        directory: c:/logs
        accept-count: 100 # Maximum queue length for incoming connection requests when all possible request processing threads are in use.
        buffered: true # Buffer output such that it is only flushed periodically.
        pattern: common # Format pattern for access logs (can be combined).
        prefix: access_log # Log file name prefix.
        rename-on-rotate: false # Defer inclusion of the date stamp in the file name until rotate time.
        request-attributes-enabled: false # Set request attributes for IP address, Hostname, protocol and port used for the request.
        rotate: true # Enable access log rotation.
        suffix: .log # Log file name suffix.
  application:
    server:
      internet-address: http://10.0.0.100:8080
      http:
        cache: # Used by the CachingResponseHeadersFilter
timeToLiveInDays: 31
data-engine:
  store:
    design-mode-limit: 400
```

Upon installation of the Knowledge Hub application, your configuration file will only include settings for writing to the Cassandra database, the connection to the Knowledge Hub database, the server port, and the SSL certificate (if this option is selected during installation).
The following table describes, in detail, the properties that may be added to this configuration file.

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SPRING</strong></td>
<td></td>
</tr>
<tr>
<td>spring.data.cassandra.enabled</td>
<td>Accepts the values <strong>true</strong> or <strong>false</strong></td>
</tr>
<tr>
<td></td>
<td>Enables (when true) or disables (when false) exports to the Library and exports of pinned data</td>
</tr>
<tr>
<td>spring.datasource.url</td>
<td>Describes the connection to the Postgres database for the Knowledge Hub service</td>
</tr>
<tr>
<td>spring.datasource.url.jdbc</td>
<td></td>
</tr>
<tr>
<td>spring.datasource.url.username</td>
<td></td>
</tr>
<tr>
<td>spring.datasource.url.password</td>
<td></td>
</tr>
<tr>
<td>spring.http.multipart.maxFileSize</td>
<td>Describes the maximum size of files that may be uploaded to the application (e.g., 2000MB)</td>
</tr>
<tr>
<td>spring.http.multipart.maxRequestSize</td>
<td></td>
</tr>
<tr>
<td><strong>SERVER</strong></td>
<td></td>
</tr>
<tr>
<td>server.port</td>
<td>Port on which the application is running</td>
</tr>
<tr>
<td>server.port.ssl.enabled</td>
<td><strong>true</strong> if HTTPS is enabled</td>
</tr>
<tr>
<td>server.port.ssl.key-store</td>
<td>Describe properties for the SSL certificate</td>
</tr>
<tr>
<td>server.port.ssl.key-store-password</td>
<td></td>
</tr>
<tr>
<td>server.port.ssl.key-password</td>
<td></td>
</tr>
<tr>
<td>server.tomcat.accesslog.enabled</td>
<td>These items describe settings for Tomcat logs:</td>
</tr>
<tr>
<td>server.tomcat.accesslog.directory</td>
<td></td>
</tr>
<tr>
<td>server.tomcat.accesslog.accept-count</td>
<td></td>
</tr>
<tr>
<td>server.tomcat.accesslog.buffered</td>
<td></td>
</tr>
<tr>
<td>server.tomcat.accesslog.pattern</td>
<td></td>
</tr>
<tr>
<td>server.tomcat.accesslog.prefix</td>
<td></td>
</tr>
<tr>
<td>server.tomcat.accesslog.rename-on-rotate</td>
<td></td>
</tr>
<tr>
<td>server.tomcat.accesslog.request-attributes-enabled</td>
<td></td>
</tr>
<tr>
<td>server.tomcat.accesslog.rotate</td>
<td></td>
</tr>
<tr>
<td>server.tomcat.accesslog.suffix</td>
<td></td>
</tr>
<tr>
<td><strong>APPLICATION</strong></td>
<td></td>
</tr>
<tr>
<td>application.server.internet-address</td>
<td>Describes the redirect URL for login to Salesforce, Google Analytics, Google Adwords (should be identical to the URL specified for ClientId and ClientSecret for these connections). etc.</td>
</tr>
<tr>
<td>application.http.cache.timeToLiveInDays</td>
<td>Describes the amount of time in days that may elapse before a data source’s cache times out</td>
</tr>
<tr>
<td>PROPERTY</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>application.data-engine.store.design-mode-limit</td>
<td>Describes the row limit to be used for data sources in Design Mode; the default value is 10K</td>
</tr>
<tr>
<td>application.data-engine.suggestion.prepare-cron</td>
<td>Describes settings for jobs that calculate suggestions based on data type and content e.g., 0 */30 * * * - jobs are run every 30 min</td>
</tr>
<tr>
<td>application.data-engine-api.url</td>
<td>URL for internal communication between the Knowledge Hub and Knowledge Hub Data Engine services (http://&lt;machine name&gt;:8081)</td>
</tr>
<tr>
<td>application.dsl.source-cleaner-cron</td>
<td>Describes settings for jobs that delete temporary objects</td>
</tr>
<tr>
<td>application.dsl.source-expiration-in-hours</td>
<td></td>
</tr>
<tr>
<td>application.dsl.temporary-item-cleaner-cron</td>
<td>Settings related to the job run cleaner</td>
</tr>
<tr>
<td>application.dsl.temporary-item-expiration-in-hours</td>
<td></td>
</tr>
<tr>
<td>application.dsl.process-run-cleaner-cron</td>
<td></td>
</tr>
<tr>
<td>application.dsl_process-run-cleaner-expiration-time</td>
<td></td>
</tr>
<tr>
<td>application.license.provider</td>
<td>Type of license provider; can be “local”, “remote”, or “hwu”</td>
</tr>
<tr>
<td>application.license.local/filepath</td>
<td>Path to license.lic file</td>
</tr>
<tr>
<td>application.license.remote.url</td>
<td>URL to the remote server</td>
</tr>
<tr>
<td>application.license.hwu.host</td>
<td>Altair License Server address. Should be written as “&lt;port&gt;@&lt;host&gt;”. Note that the URL to the Altair License Server should be set as an environment variable.</td>
</tr>
<tr>
<td>application.license.hwu.checker-cron</td>
<td>Schedule to execute remote license pool check (e.g., 00/5 * * * *)</td>
</tr>
<tr>
<td>application.license.hwu.group</td>
<td>Name of group on Altair License Server (e.g., $(COMPUTERNAME)). Note that this property should also be set as an environment variable.</td>
</tr>
<tr>
<td>application.license.hwu.log.enabled</td>
<td>Enable (true) or disable (false) hwu logging</td>
</tr>
<tr>
<td>application.license.hwu.log.level</td>
<td>Level of hwu logging (e.g., info)</td>
</tr>
<tr>
<td>application.license.hwu.logfacility</td>
<td>Type of output (e.g., stderr)</td>
</tr>
<tr>
<td>application.license.filepath</td>
<td>Describes the path to the Knowledge Hub license</td>
</tr>
<tr>
<td>application.io.connection.disabled</td>
<td>Specifies which connection types to disable (hide) in Knowledge Hub for all users</td>
</tr>
<tr>
<td>application.io.appDataFolder</td>
<td>Describes the path to the application’s internal storage (i.e., File Library; default: ${ProgramData}/Datawatch/DNS</td>
</tr>
<tr>
<td>PROPERTY</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>application.io.writer.cognos.http-client.timeout</td>
<td>Describes the time in seconds that may elapse before connections to IBM Cognos Analytics time out e.g., 600</td>
</tr>
<tr>
<td>application.io.reader.preview.limit</td>
<td>1000 – row limit for previewing data sources</td>
</tr>
<tr>
<td>application.security.authentication.xauth.token ValidityInSeconds</td>
<td>Describes how many seconds should elapse before a user times out e.g., 1800</td>
</tr>
<tr>
<td>application.schedules.monitoring.intervalInMinutes</td>
<td>Number of minutes that must elapse before the next monitoring operation should be executed in a monitoring schedule</td>
</tr>
<tr>
<td>application.security.authentication.failed-attempt-min-delay-sec</td>
<td>Delay after the first failed login attempt e.g., 8</td>
</tr>
<tr>
<td>application.security.authentication.failed-attempt-max-delay-sec</td>
<td>Maximum delay time after a failed login attempt e.g. 600</td>
</tr>
</tbody>
</table>

Knowledge Hub Data Engine

Settings for the Knowledge Hub Data Engine service are specified in C:\Program Files\Datawatch Monarch Swarm\dp\application-prod.yml. A snippet of the config file is provided below:

```yaml
spring:
data:
cassandra:
enabled: true
datasource:
url: jdbc:postgresql://localhost:5433/dataengine
logging:
file: logs/app.log
logback:
loglevel: INFO
server:
port: 8881
ssl:
enabled: false
application:
data-engine:
suggestion:
rank-threshold: 0.2 # default ranking thresholds, min value
store:
export-data-await-timeout-in-sec: 3600
design-mode-limit: 10000
globalRowLimit: 1000000
column-limit: 100
distinct-value-limit: 250
```
The following table describes, in detail, the properties that may be added to this configuration file.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SPRING</strong></td>
<td></td>
</tr>
<tr>
<td>spring.data.cassandra.enabled</td>
<td>Accepts the values <strong>true</strong> or <strong>false</strong>. Enables (when true) or disables (when false) exports to the Library and exports of pinned data.</td>
</tr>
<tr>
<td>spring.datasource.url.jdbc</td>
<td>Describes the connection to the Postgres database for the Knowledge Hub service.</td>
</tr>
<tr>
<td>spring.datasource.url.username</td>
<td></td>
</tr>
<tr>
<td>spring.datasource.url.password</td>
<td></td>
</tr>
<tr>
<td><strong>LOGGING</strong></td>
<td></td>
</tr>
<tr>
<td>logging.file</td>
<td>full path to Data Engine service log file</td>
</tr>
<tr>
<td><strong>LOGBACK</strong></td>
<td></td>
</tr>
<tr>
<td>logback.loglevel</td>
<td>Logging level of the Data Engine service log file</td>
</tr>
<tr>
<td><strong>SERVER</strong></td>
<td></td>
</tr>
<tr>
<td>server.port</td>
<td>8081 – port on which the Data Engine is running</td>
</tr>
<tr>
<td>server.port.ssl.enabled</td>
<td><strong>true</strong> if HTTPS is used.</td>
</tr>
<tr>
<td>server.port.ssl.key-store</td>
<td></td>
</tr>
<tr>
<td>server.port.ssl.key-store-password</td>
<td></td>
</tr>
<tr>
<td><strong>APPLICATION</strong></td>
<td></td>
</tr>
<tr>
<td>application.data-engine.suggestion.rank-threshold</td>
<td>Describes settings for suggestions based on data type and content; shows minimum rank for retrieving and sorting suggestions.</td>
</tr>
<tr>
<td>application.data-engine.store.statistics-await-timeout</td>
<td>Time to wait before statistics requests time out e.g., 60s</td>
</tr>
<tr>
<td>application.data-engine.store.design-mode-limit</td>
<td>Describes the row limit to be used for data sources in Design Mode; the default value is 10K</td>
</tr>
<tr>
<td>application.data-engine.store.globalRowLimit</td>
<td>Row limit applied when the Design Mode limit is disabled e.g., 5000</td>
</tr>
<tr>
<td>application.data-engine.store.column-limit</td>
<td>100 - column limit after Pivot and Transpose.</td>
</tr>
<tr>
<td>application.data-engine.store.distinct-value-limit</td>
<td>250 - number of displayed distinct values limit</td>
</tr>
<tr>
<td>PARAMETER</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>application.data-engine.store.limit-data-nodes</td>
<td>Enables or disables limit to count of rows in all data nodes e.g., true (enabled); false (disabled)</td>
</tr>
<tr>
<td>application.data-engine.store.export-data-await-timeout-in-sec</td>
<td>3600 - export timeout</td>
</tr>
<tr>
<td>application.io.writer.cognos.http-client.timeout</td>
<td>600 - timeout for connection to IBM Cognos Analytics</td>
</tr>
<tr>
<td>application.server.internet-address</td>
<td>Redirect URL for logins to Salesforce, Microsoft Sharepoint, Google Analytics, Google BigQuery, Google Adwords, Google Drive (redirect url should be specified for ClientId and ClientSecret for Google connections).</td>
</tr>
<tr>
<td>application.security.authentication.xauth.secret</td>
<td>security token, should be equal to all other security tokens in all other application config files</td>
</tr>
<tr>
<td>application.core-api.url</td>
<td>address for internal communication with the Knowledge Hub service</td>
</tr>
</tbody>
</table>

**READER**

| application.io.reader.preview.limit                                     | 1000 - row limit for preview data sources.                                                                                                                                   |

**JDBC**

<p>| application.io.reader.jdbc.fetch-size                                   | Describes the number of rows to fetch for a query to a database using JDBC drivers, e.g., 200                                                                                  |
| application.io.reader.jdbc.timeout-in-sec                               | Describes the time in seconds that may elapse before connections to JDBC drivers time out e.g., 60                                                                          |
| application.io.reader.jdbc.driver.*                                     | Configuration settings for JDBC drivers                                                                                                                                 |
| application.io.reader.jdbc.driver.default.loginTimeout                  | Describes the time in seconds that may elapse before connections to JDBC drivers time out after login e.g., 60                                                            |
| application.io.reader.jdbc.driver.default.socketTimeout                 | Describes the time in seconds that may elapse before a socket timeout occurs when using connections to JDBC drivers e.g., 60                                                  |
| application.io.reader.jdbc.driver.cdata.jdbc.all.timeout               | Describes the time in seconds that may elapse before all connections to JDBC drivers time out e.g., 60                                                                 |
| application.io.reader.jdbc.driver.com.mysql.jdbc.Driver.useCursorFetch  | Settings for mySQL JDBC driver                                                                                                                                               |
| application.io.reader.jdbc.driver.com.mysql.jdbc.Driver.loginTimeout    |                                                                                                                                                                               |</p>
<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>application.io.reader.jdbc.driver.com.mysql.jdbc.Driver.socketTimeout</td>
<td></td>
</tr>
<tr>
<td>application.io.reader.jdbc.driver.oracle.jdbc.OracleDriver.oracle.net.</td>
<td>Settings for Oracle JDBC driver</td>
</tr>
<tr>
<td>CONNECT_TIMEOUT</td>
<td></td>
</tr>
<tr>
<td>application.io.reader.jdbc.driver.oracle.jdbc.OracleDriver.oracle.jdbc.</td>
<td></td>
</tr>
<tr>
<td>ReadTimeout</td>
<td></td>
</tr>
<tr>
<td>application.io.reader.jdbc.driver.com.facebook.presto.jdbc.PrestoDriver</td>
<td>Settings for Presto JDBC driver</td>
</tr>
<tr>
<td>presto.jdbc.PrestoDriver.SSL</td>
<td></td>
</tr>
<tr>
<td>presto.jdbc.PrestoDriver.password</td>
<td></td>
</tr>
<tr>
<td>application.io.reader.jdbc.driver.org.apache.hive.jdbc.HiveDriver.java.</td>
<td>Settings for Apache Hive JDBC driver</td>
</tr>
<tr>
<td>security.krb5.conf</td>
<td></td>
</tr>
<tr>
<td>security.auth.login.config</td>
<td></td>
</tr>
<tr>
<td>report-text-view-max-cache-in-mb</td>
<td>This option sets the limit in megabytes for storing the converted reports.</td>
</tr>
<tr>
<td>report-text-view-max-cache-count</td>
<td>This option sets the limit in counts for storing the converted reports.</td>
</tr>
<tr>
<td>report-text-view-number-of-pages</td>
<td>If the page number of THE report exceeds this setting, then conversion</td>
</tr>
<tr>
<td></td>
<td>option converts from PDF reports to TXT report</td>
</tr>
</tbody>
</table>
Knowledge Hub ML and Spark

The config file for the Knowledge Hub ML and Spark services is typically installed in C:\Program Files\Datawatch Monarch Swarm\ml\social-application.yml.

```yaml
server:
  jwt:
    secret: secret as in all other configs
  ssl:
    enabled: false
    keystore.password: secret
    keystore.path: ..\\certificate\localhost.jks
spark:
  app:
    name: DW Swarm Decision Engine
    schedule: 0 0 0 1/* ? *
    scheduling:
      enabled: true
  driver:
    memory: 4g

database:
  cassandra:
    connection:
      attempts:
        amount: 8
      wait:
        time:
          seconds: 30
logging:
  loglevel: INFO
  logfilepath: ${TEMP}/MonarchSwarm/Logs/ml-app.log
```

This table describes, in detail, the parameters that may be added to the configuration file.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SERVER</strong></td>
<td></td>
</tr>
<tr>
<td>server.jwt.secret</td>
<td>A security token that should be identical to all other security token indicated in all other application config files (e.g., application.security.authentication.xauth.secret in other config files)</td>
</tr>
<tr>
<td>server.port.ssl.enabled</td>
<td>true if HTTPS is used. Describes parameters for the SSL certificate.</td>
</tr>
<tr>
<td>server.port.ssl.keystore-password</td>
<td></td>
</tr>
<tr>
<td>server.port.ssl.keystore.path</td>
<td></td>
</tr>
<tr>
<td>PARAMETER</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>SPARK</strong></td>
<td></td>
</tr>
<tr>
<td>spark.app.name</td>
<td>Spark application name</td>
</tr>
<tr>
<td>spark.app.schedule</td>
<td>Cron for Spark job for calculating suggestions (e.g., 0 0/20 * 1/1 * ? *)</td>
</tr>
<tr>
<td>spark.app.scheduling.enabled</td>
<td><strong>False</strong> if only a single run is applied</td>
</tr>
<tr>
<td>spark.driver.memory</td>
<td>Allocated memory size for Spark service</td>
</tr>
<tr>
<td><strong>DATABASE</strong></td>
<td></td>
</tr>
<tr>
<td>database.cassandra.connection.attempts.amount</td>
<td>Number of attempts to connect to the Cassandra database</td>
</tr>
<tr>
<td>database.cassandra.connection.wait.time.seconds</td>
<td>Number of second for each attempt to connect to the Cassandra database</td>
</tr>
<tr>
<td><strong>LOGGING</strong></td>
<td></td>
</tr>
<tr>
<td>logging.loglevel</td>
<td>Logging level</td>
</tr>
<tr>
<td>logging.logfilepath</td>
<td>Full path to the ML and Spark services log file (e.g., C:\Windows\Temp\MonarchSwarm\Logs\ml-app.log)</td>
</tr>
<tr>
<td><strong>NEXT-CHANGE</strong></td>
<td></td>
</tr>
<tr>
<td>next-change.min-changes</td>
<td>Minimum number of actions in sequence to generate suggestions</td>
</tr>
</tbody>
</table>
Upgrading Knowledge Hub

Upgrading the Knowledge Hub Application

To gain access to Knowledge Hub’s newer features, you must upgrade the application. Note that cached data sets are reset when upgrades are performed. Statistics are not reset.

To upgrade your Knowledge Hub installation, simply run the installer of the newer version of the application. While no uninstallation of the previous version is necessary, the Cassandra, Knowledge Hub, Knowledge Hub Data Engine, Knowledge Hub ML, and Knowledge Hub Spark services must be stopped prior to the upgrade.

Steps:
1. Double-click on the application installer to run it.
2. Read the license agreement that displays in the next screen and, if you agree with its terms, select the button for I accept the agreement. Click Next when you are finished.

The Database Configuration screen displays.
3. In the fields provided, enter the **PostgreSQL administrator login** and **password** that were provided when the older version of Knowledge Hub was installed.

4. Click **Next** when you are finished.

   The **Select Components** screen displays.

5. If you wish to make changes to the components you had previously installed, you can do so now. Otherwise, simply click **Next**.
The *Ready to Install* screen displays.

6. Click **Install**.

The installation begins.
7. After installation, the following screen displays.

8. Click Finish.

The Knowledge Hub login page displays.

9. If you wish to implement HyperWorks Units-based licensing, stop all Knowledge Hub services. Create a new environment variable with the name ALTAIR_LICENSE_PATH and set its value to "<license server port>@<license server host>" (e.g., 6200@100.0.0.100). Restart all services and log in to Knowledge Hub as usual.

Upgrading Java Runtime Environment

The Java Runtime Environment (JRE) version in the server in which Knowledge Hub has been installed can be upgraded. To do so, download the new JRE version from the official Oracle site, uninstall the old JRE version, and then install the new one. Proceed with running the installer or upgrading Knowledge Hub.

Alternatively, the steps below describe how to upgrade Java Runtime Environment if the application is not going to be upgraded or reinstalled.

Steps:

1. Stop all of the related Knowledge Hub services.
2. Go to Control Panel\System and Security\System and select Advanced system settings.
3. In the System Properties dialog that displays, click Environment Variables.
4. Under System variables, change the Java path in Path to the correct value.
For example, if the current path reads "C:\Program Files\Java\jre1.8.0_161\bin" and your JRE was upgraded to jre1.8.0_181, change the path to "C:\Program Files\Java\jre1.8.0_181\bin" and then click OK on the Edit System Variable dialog to close it.

5. Check that the new path is applied.

6. Click OK on the Environment Variables and System Properties dialogs to close them.

7. In C:\Program Files\PostgreSQL\9.6\pgAdmin 4\bin, double-click on the pgAdmin4 application to launch it.
8. Connect to the **PostgreSQL 9.6** server using the default user "postgres" and password "postgres."

9. Locate the database newserver_dataengine > Schemas > public.

10. Open the Query Tool by right-clicking on the schema **public** and then selecting **Query Tool** from the options that display.
11. Execute the following script:

   SET pljava.libjvm_location TO 'path';
   
   ALTER DATABASE newserver_dataengine SET pljava.libjvm_location FROM CURRENT;

   where path is the path to the jvm.dll file of the upgraded JRE (e.g., C:\Program Files\Java\jre1.8.0_181\bin\server\jvm.dll),

12. Execute this script by clicking the Execute/Refresh (F5) icon.

13. Close pgAdmin4 and restart the Knowledge Hub service.

**NOTE**

If your Knowledge Hub application was installed using the HTTPS protocol and Java Runtime Environment was upgraded, you must generate a new security certificate. Click [here](#) for instructions on how to obtain this certificate and configure client machines.

**IMPORTANT:** When upgrading to Knowledge Hub version 2.3.x from a lower version of the application, the JRE upgrade procedure must be completed prior to the application upgrade. If not done in this order, the installer may encounter issues when migrating data from PostgreSQL 9.6 to 11.
Troubleshooting Knowledge Hub

The items in this section describe troubleshooting measures you may need to take when working with Knowledge Hub.

Uninstalling the Application

Knowledge Hub may be uninstalled prior to upgrading it.

Steps:

1. In the folder Control Panel > Programs > Uninstall a program, locate the Knowledge Hub application, right-click on it, and then select Uninstall from the menu that displays.

2. Click Yes on the message box that displays.

   Knowledge Hub is uninstalled.

3. Uninstall the Datawatch JDBC drivers and PostgreSQL.

   Note that other applications besides Knowledge Hub may require PostgreSQL to run correctly. Consult your system administrator before uninstalling the program. Information on how to completely remove PostgreSQL from your server may be found online.

4. Restart your computer.

   A newer version of Knowledge Hub may now be installed.
Starting the Service

In some instances, Knowledge Hub may not be accessed from any browser because the service has not been started. This may occur when the operating system of the machine in which the application is installed is updated or restarted.

If the Knowledge Hub application does not start automatically, you may need to start it manually.

Steps:


2. In Services window that displays, locate the Datawatch Monarch Swarm item and then click Start in the left-hand pane of the Services list.

3. Repeat Step 2 for the Knowledge Hub Data Engine, Knowledge Hub ML, and Knowledge Hub Spark services.

NOTE

You can also select the item, right-click on your mouse, and then select Start from the options that display.

The Knowledge Hub services may be started in any order.
4. Close the Services window.

You may need to wait a few minutes to allow these services to restart completely.

The Knowledge Hub login page can now be loaded as usual.

Starting the Cassandra Service

There may be instances when the Cassandra service fails to start automatically (for example, after an OS restart). This may occur because of a corrupted commitlogs file. The following steps describe how to start the Cassandra service when the issue is a corrupted commitlogs file.

Steps:

1. Open the Cassandra log file. This file is usually located in C:\Program Files\Datawatch Monarch Swarm\cassandra\logs\system.log.

2. Check the name of the corrupted commitlog file with the latest error.

In the example below, the corrupted commitlog file is C:\Program Files\Datawatch Monarch Swarm\cassandra\data\commitlog\CommitLog-6-1509613026245.log.
3. Delete this file from the file system indicated.
4. Start the Cassandra service.

Back up the Knowledge Hub PostgreSQL Database

Backing up your Knowledge Hub databases is necessary to ensure that all of your data, workspace definitions, security settings, and the like are preserved when PostgreSQL or the application itself is upgraded or reinstalled.

When Knowledge Hub is installed, PostgreSQL 9.6 and its management tool, pgAdmin4, are concurrently installed. This tool can be used to back up and restore your database. The PostgreSQL database includes information on, among others, users, data source/workspace/connection metadata, and permissions.

**NOTE**

The binary path to the directory containing the PostgreSQL utility programs (e.g., pg_dump, pg_restore, etc.) must be specified in **File > Preferences > Paths > Binary paths** before you can back up a database. Contact your system administrator if you need help doing so. More information on this setting may be found [here](#).

**Steps:**

1. In **C:\Program Files\PostgreSQL\9.6\pgAdmin 4\bin**, double-click on the pgAdmin4 application to launch it.
The pgAdmin4 user interface displays.

2. Expand the Servers node and then double-click **PostgreSQL 9.6**.

A login dialog displays.

3. Enter the password for the user **postgres** and then click **OK**.

You are connected to the PostgreSQL server, and several items display below it.
4. Expand the **Databases** node and then select the Knowledge Hub database. Right-click on it and then select **Backup** from the options that display.

The **Backup** dialog displays.

5. Provide the information required in the **General** tab of this dialog and then click on the **Dump options** tab.
NOTE

Save the information you provide in the General tab of the Backup dialog. You will need to provide this information when restoring the database at a later time.

6. Move the switches in the Sections box to specify which portions of the database should be backed up.

7. Move the switches in the Type of Objects box to specify which database components to back up.

8. Move the switches in the Do not save box to specify which database components to exclude from the backup.

9. Move the switches in the rest of the field boxes provided in the tab to specify other settings for your backup.

10. Click Backup at the bottom of the dialog when you are finished. Otherwise, click Cancel to abort the backup operation.

A popup dialog similar to that shown below displays to indicate successful or unsuccessful backup.
Details of the backup operation, whether successful or not, may be viewed in the **Process Watcher**, which is launched by clicking on the **Click here for details** link provided in the popup.

By default, your backup will be located in `C:\Program Files\PostgreSQL\9.6\pgAdmin 4\bin`.

More information on how to backup databases in PostgreSQL may be found [here](#).

## Restoring the Knowledge Hub PostgreSQL Database

When PostgreSQL or Knowledge Hub is upgraded or reinstalled, you may need to restore a previously backed-up Knowledge Hub database to continue working with your data.

The pgAdmin4 tool is also used to restore Knowledge Hub databases.

### Steps:

1. Run the **pgAdmin4** tool and log into the **PostgreSQL 9.6** server.
2. Create a new Knowledge Hub database. If this database already exists, select it.
3. Right-click on this database and, from the options that display, select **Restore**.

   The **Restore** dialog displays.

4. Provide the information required by each of the fields in the **General** tab of this dialog. This information should match the details you specified in the **General** tab of the **Backup** dialog when you backed up your database.

5. Click the **Restore options** tab.
6. Use the switches provided in each of the field boxes in this tab to specify how your database should be restored.

7. Click **Restore** at the bottom of the dialog when you are finished. Otherwise, click **Cancel** to abort the restore operation.

   A popup window displays to indicate successful or unsuccessful restoration.

   Details of the restore operation, whether successful or not, may be viewed in the **Process Watcher**, which is launched by clicking on the **Click here for details** link provided in the popup.

   More information on how to restore databases in PostgreSQL may be found [here](#).

---

**Backing up the Cassandra Database**

The following objects are created as separate tables in the Cassandra database under the keyspace **newserver**.

- Tables exported to the Knowledge Hub library from the application
- Tables exported from Data Prep Studio to Knowledge Hub
- Workspaces with pinned tables saved to Knowledge Hub

The steps below describe how to save snapshots of these tables for back up.

**Steps:**

1. Set the JAVA_HOME path by going to **Control Panel**\**System and Security**\**System** and then selecting **Advanced** system settings.

2. In the **System Properties** dialog that displays, click **Environment Variables**.

3. In the **Environment Variables** dialog, click **New...** under System variables, enter **JAVA_HOME** as the variable name and then enter the path to your Java application in the **Variable value** field. Click **OK** when you are done and then **OK** in the **System Properties** dialog.

4. Launch **cmd.exe**.

5. Go to the bin directory of Cassandra.

   **Example:** `C:\Program Files\Datawatch Monarch Swarm\cassandra\bin`

6. Run the following command to create the snapshot for one keyspace:

   ```bash
   nodetool -h host -p port snapshot keyspace_name
   ```

   **Example:** `nodetool -h localhost -p 7199 snapshot newserver`
NOTE

Knowledge Hub creates two keyspaces in the Cassandra database: **newserver** and **datawatch**. Both keyspaces should be backed up.

The files created are saved in the snapshot directory, usually in `C:\{Knowledge Hub Home}\cassandra\data\data\{keyspace name}\{table name}\snapshots`

Folder example:

```
C:\Program Files\Datawatch Monarch Swarm\cassandra\data\data\newserver\data_scheme-
3486829030af11e7a545dd2ebe24b125\snapshots\1493962486390
```

BACKING UP THE CASSANDRA DATABASE FOR KNOWLEDGE HUB VERSIONS 2.0 AND EARLIER

The steps described above may not produce the desired results if you are working with Knowledge Hub versions prior to version 2.1 or upgraded and older version to 2.1 and the installation path for Cassandra contains spaces. This issue originates from the default Cassandra configuration file.

In this case, the following steps should be done before snapshot creation:

**Steps:**

1. Open Services and stop the Cassandra service.

2. Open the cassandra directory and update Line 380 of the default config for Cassandra database (usually in `C:\Program Files\Datawatch Monarch Swarm\cassandra\conf\cassandra-env.ps1`) as follows:

   **Default:**
   ```
   $env:JVM_OPTS = "$env:JVM_OPTS -
   XX:CompileCommandFile=$env:CASSANDRA_CONF\hotspot_compiler"
   ```

   **Update to:**
   ```
   $env:JVM_OPTS = "$env:JVM_OPTS -
   XX:CompileCommandFile="$env:CASSANDRA_CONF\hotspot_compiler"
   ```

3. Open command prompt with Administrator rights, go to the bin directory of Cassandra (usually in `C:\Program Files\Datawatch Monarch Swarm\cassandra\bin`) and then run the following command:

   `cassandra -install`

4. Restart the Cassandra service.
Restoring the Cassandra Database

You can restore your Cassandra database to repair corrupted tables or create a completely new node. Note that the steps outlined in this section assume that you have properly backed up your Cassandra database and that all of the necessary snapshots are saved in the snapshot directory.

To restore the Cassandra database in the case of corrupted data, the following steps are taken.

Steps:
1. Open cmd.exe.
2. Go to C:\Program Files\Datawatch Monarch Swarm\cassandra\bin.
3. Run nodetool drain.
4. Shut down the Cassandra and Swarm services.
5. Delete all of the files in the Knowledge Hub commitlog directory (in C:\Program Files\Datawatch Monarch Swarm\cassandra\data\commitlog).
6. Delete all files (except folders) in the directory C:\Program Files\Datawatch Monarch Swarm\cassandra\data\data\{keyspace name}\{table name}\.

   Directory example: C:\Program Files\Datawatch Monarch Swarm\cassandra\data\data\newserver\ds_10015-e36e6f90349011e7858edcb96aa2c5e0

   Note that Step 6 must be repeated for all corrupted tables.
7. Go to the snapshot directory and copy all of the contents (per table) to the matching data_directories from which you deleted DB files in the previous step.

   Step 7 must be repeated for all tables to be restored.
8. Restart the Cassandra and Knowledge Hub services.
10. Restart all of the Knowledge Hub services.

Follow the steps below to restore create a completely new node using saved Cassandra table snapshots.

Steps:
1. Set the Python path by going to Control Panel > System and Security > System > Advanced Settings > Environment Variables.
2. Add C:\Program Files\Datawatch Monarch Swarm to the PATH variable.
3. Go to C:\Program Files\Datawatch Monarch Swarm\cassandra\bin.
4. Copy all of the schema.cql files generated from backup to the bin directory.
5. Recreate the schema. To do so, type `cqlsh -f schema.cql` into the command prompt and press Enter on your keyboard.

6. Copy the backup files to the corresponding directories. Note that this step must be performed for all directories to be restored.

7. In command prompt, run `nodetool refresh newserver <tablename>`.
   
   E.g.: `nodetool refresh newserver ds_10015`

8. Restart all of the Knowledge Hub services.

Product Support

Product support for Knowledge Hub may be obtained here.