Hyland Preferences Service

Installation Guide

Version: 3.0.x

Written by: Documentation Team, R&D
Date: August 2023

Hyland

Documentation Notice

Table of Contents

Overview

This document assumes that you are installing the Hyland Preferences Service database for the first time. Before you install, verify that your system meets the requirements listed in the *Hyland Preferences Service Technical Specifications*.

Download the Hyland Preferences Service Database files

To obtain Hyland product installation files, contact the Hyland Software Technical Support group. For a list of Technical Support phone numbers, go to hyland.com/pswtscontact.

About the USRPREFSVC account

The USRPREFSVC database user is the user that owns all the Hyland Preferences Service database objects and is used by the service to connect to the database. The SQL creation scripts attempt to create the USRPREFSVC account if it does not exist. Changing the password of this user is highly recommended.

Setup a Microsoft SQL Server Hyland Preferences Service database

Follow the procedures in the following sections to create the Hyland Preferences Service database in Microsoft SQL Server.

Note It is not necessary to create a database if one already exists that is used for other Hyland services, and fits the specifications required by the Hyland Preferences Service.

Connect to your SQL server

Complete the following steps to connect to your SQL server.

1. On a computer with access to the server, open **SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, complete the following substeps.
    1. In the **Server Type** list, select the server type.
    2. In the **Server Name** list, select the name of the server.
    3. In the **Authentication** list, select the authentication type.
    4. Optional. Enter the **User name** and **Password** credentials if needed.
3. Click **Connect**.

Verify Microsoft SQL server instance properties

Complete the following steps to verify Microsoft SQL server instance properties.

1. In **SQL Server Management Studio**, in the **Object Explorer** pane, right-click your server and then click **Properties**.

    Note Typically, your server is the first item listed in the tree.
2. In the **Server Properties** dialog box, under **Select a Page**, click **Security**.

3. Under **Server Authentication**, select **SQL Server and Windows Authentication Mode**, and then click **OK.**
4. If prompted, click **Yes** to restart the server.

   Important Restarting the server shuts down and starts up all databases on this instance.

(Optional) Create a new database

If a database does not already exist that meets the needs of the Hyland Preferences Service, you can choose to create a new one.

Required database collation

Hyland Preferences Service requires case-insensitive collation. **Note** Case insensitive collations will include CI in their name.

1. In **SQL Server Management Studio**, in the **Object Explorer** pane, expand the server node.
2. Right-click **Databases** and select **New Database**.
3. In the **Database name** box, enter the name of the database according to the naming conventions of your organization. For example, **UserPreferences**.
4. In the **Select a page** pane on the left, click **Options**.
5. Change the **Collation** property to an appropriate language. For example, **Latin1_General_100_CI_AS**.
6. Set other options according to your company's best practices.
7. Click **OK** to complete creation**.**

Create the Hyland Preferences Service schema objects

The **PreferencesServiceSS.sql** script creates the USRPREFSVC user and schema, and tables required by the Hyland Preferences Service.

1. Open **PreferencesServiceSS.sql** in **SQL Server Management Studio**.
2. Change the selected database to the database where you would like to create the service tables.
3. Click **Execute**.

Important Change the default password of the USRPREFSVC user by editing the SQL script prior to executing, or by updating the user after creation.

Setup an Oracle Hyland Preferences Service database

Follow the procedures in the following sections in order to create the Hyland Preferences Service database in Oracle.

Note It is not necessary to create a database if one already exists that is used for other Hyland services, and fits the specifications required by the Hyland Preferences Service.

(Optional) Create a new database

If a database does not already exist that meets the needs of the Hyland Preferences Service, you can choose to create a new one.

Create an Oracle database using your organization's best practices with regard to multitenancy, security, storage and recovery. There are no special database options required. Feel free to use your preferred program to create Oracle databases.

For example, you can follow the procedures in this section to create a database using **Oracle Database Configuration Assistant**, but the use of this program is not required. **Note** These steps were documented using an **Oracle 18c** version of **DBCA**.

1. Using the **Oracle Database Configuration Assistant** (DBCA), create a new database and name it according to the naming conventions for your organization. For example, **UserPreferences**.
2. On the **Storage Options** page, choose the appropriate database storage attributes in accordance with your organizations best practices.
3. On the **Fast Recovery Options** page, choose the appropriate recovery options in accordance with your organizations best practices.
4. On the **Network Configuration** page, select an existing listener or create a new listener.
5. On the **Database Options** page, you can deselect all the components or select the specific components that you might want to use. The **User Preference** database does not require any of the other standard database components.
6. On the **Configuration Options** page complete the following substeps.
    1. On the **Memory** tab allocate 2GB or more for the SGA size and 128MB or more for the PGA. Review the memory advisory statistics after deployment and adjust memory settings as necessary.
    2. On the **Sizing** tab use the default 8K block size. The number of processes will be relatively low for this environment so you can start at 150 and increase later if necessary.
    3. On the **Character sets** tab you can choose to use the Unicode or OS character set. The User Preferences schema will be using variable-length Unicode datatypes (nvarchar2).
    4. On the **Connection mode** tab select the Dedicated server mode. The number of connections to this database will be relatively low.
7. On the **Management Options** page, follow the best practices set by your organization for configuring Enterprise Manager (EM).
8. On the **Database User Credentials** page, follow the best practices set by your organization for the administrative accounts and their passwords.
9. On the **Database Creation Option** page, make any additional advanced configuration option changes to the initialization parameters and storage locations if necessary.
10. Review the **Summary** and **Finish** the database creation.

Create the Hyland Preferences Service schema objects

The **PreferencesServiceORA.sql** script creates the **USRPREFSVC** user and schema, and tables required by the Hyland Preferences Service.

1. Connect to the Hyland Preferences Service database as SYS or another user that contains the Database Administrator role.
2. Execute the **PreferencesServiceORA.sql** script.

Important Change the default password of the USRPREFSVC user by editing the SQL script prior to executing, or by updating the user after creation.

Setup an PostgreSQL Hyland Preferences Service database

Follow the procedures in the following sections in order to create the Hyland Preferences Service database in PostgreSQL.

Note It is not necessary to create a database if one already exists that is used for other Hyland services, and fits the specifications required by the Hyland Preferences Service.

(Optional) Create a new database

1. Open **pgAdmin 4**.
2. Connect to the appropriate PostgreSQL instance using the postgres login role or a login role with superuse privileges.
3. Right-click on **Databases**, select **Create** and then select **Database**. The system opens the **Create – Database** dialog box.
4. On the **General** tab, complete the following substeps.
    1. In the **Database** field, enter a name for the database.
    2. In the **Owner** field, enter the owner for the database.
    3. In the **Comment** field, enter any additional notes you would like to document for this database.
5. On the **Definition** tab, in the **Encoding** list, select **UTF8** and accept the default values for the remaining lists and fields.
6. Accept the default values for the remaining tabs, **Security**, **Parameter**, **Advanced**, and **SQL**.
7. Click **Save**.

Create the Hyland Preferences Service schema objects

The **PreferencesServicePG.sql** script creates the **USRPREFSVC** login role and schema, and the default tables required by the Hyland Preferences Service. You can also run the **PreferenceServicePG_wTS.sql** script instead which allows you to create a **usrprefsvc** tablespace to physically separate the usrprefsvc tables and indexes if desired.

1. Open **pgAdmin 4**.
2. Connect to the appropriate PostgreSQL instance using the postgres login role or a login role with superuse privileges.
3. Right-click on the appropriate database and select **PSQL Tool**.
4. Execute the appropriate script, **PreferencesServicePG.sql** or **PreferenceServicePG_wTS.sql**.

    Example:

```
usrprefsvc=> \i 'C:/PreferencesServicePG.sql'
```

Install Hyland Preferences Service as a service

Follow the procedures in the following sections to install the Hyland Preferences Service as service on 64-bit Windows or Linux.

Important This document assumes you are installing Hyland Preferences Service for the first time or that you have no earlier versions running on the computer. If an earlier version exists, backup any configuration files and remove the version.

Download the Hyland Preferences Service installation files

To obtain Hyland product installation files, contact the Hyland Software Technical Support group. For a list of Technical Support phone numbers, go to hyland.com/pswtscontact.

The required files will vary depending on which setup method and operating system are in use. Instructions in this document will specify which files are required.

Install as a Windows service

1. Download **Hyland.Preferences.Service.WindowsService.msi**.
2. Run the installer as administrator.
3. On the **Destination Folder** page, select an installation location and make note of it.
4. Click **Next.**
5. Click **Install.**
6. Configure the service. Refer to the **Configure Hyland Preferences Service** section of this document.
7. Open the **Windows Start** menu.
8. Search for and open **Services.msc**.
9. Right-click the **Hyland Preferences Service** and click **Restart**.

Install as a Windows service unattended

The **Hyland.Preferences.Service.WindowsService.msi** installer can be ran from command line using **msiexec** on Windows. This can be especially helpful in automated environment deployment.

The following command is an example of using **msiexec** to silently install Hyland Preferences Service.

```
msiexec /i <path-to-msi> /q <additional-options>
```

Below is a list of recommended additional options to set. Refer to **msiexec** documentation for additional options.

Argument	Description	Default	Example
L*V	This value is optional. It defines where to save installation logs. Note that the logs parent directory is not created automatically.		/L*V “c:/logs/preferences-service-install.txt”
INSTALLFOLDER	The installation directory.	c:/Program Files/Hyland/hyland-preferences-service	INSTALLFOLDER=”c:/Hyland/Hyland Preferences Service”

Configure the service

After the silent installation is complete, complete the following steps to configure the service.

1. Configure the service. Refer to the **Configure Hyland Preferences Service** section of this document.
2. Open the **Windows Start** menu.
3. Search for and open **Services.msc**.
4. Right-click the **Hyland Preferences Service** and click **Restart**.

Install as an IIS service

You can run Hyland Preferences Service as an IIS web application. Follow the procedures in this section to install the service as an IIS service.

Note Setup and configure IIS prior to installing the service.

Install required prerequisites

To allow Microsoft Windows to automatically install the ASP.NET Core Runtime 6.0 Windows Hosting Bundle, complete the following steps.

1. On your device, open the **Control Panel**, click **Programs** and then **Programs and Features**.
2. Click **Turn Windows features on or off**.
3. In the **Windows Features** dialog box, select **Internet Information Services Hostable Web Core**, and then click **OK**. The system downloads and installs the required prerequisites.

   Note If this feature is not available, download and install the ASP.NET Core Runtime 6.0 Windows Hosting Bundle from Microsoft.

Install the service

Complete the following steps to install the service.

1. Download **Hyland.Preferences.Service.msi.**
2. Run the installer as administrator.

   Note Any required missing prerequisites are checked.
3. On the **IIS Settings** page, select the desired **Web Site**, **Application Name**, and **Application Pool**.

4. Click **Next**.
5. On the **Destination Folder** page, select an installation location and make note of it.
6. Click **Next**.
7. Click **Install**.
8. Configure the service. Refer to the **Configure Hyland Preferences Service** section of this document.
9. Open the **Windows Start** menu**.**
10. Search for and open **Internet Information Services (IIS) Manager**.
11. In the **Connections** pane, expand **Sites**.
12. Click the site that matches the **Web Site** chosen during installation.
13. In the **Manage Website** pane, select **Restart**.

Install as a Linux service

You can run Hyland Preferences Service as a **systemd service** on Linux. Follow the steps in this section to setup and run the service.

Note .NET Core runtime dependencies are required to be installed prior to running the service. Refer to Microsoft's documentation to determine what prerequisites are required.

1. Download **hyland-preferences-service-<version>.gz**.
2. Extract the archive. The extracted folder will contain a **hyland-preferences-service** folder and **setup.sh**.

   ```
   tar –xzvf hyland-preferences-service-<version>.gz
   ```

3. Open **shell** as a user with elevated privileges.
4. Change the current directory in **shell** to the extracted folder. cd <extracted-directory>
5. Make the **setup.sh** script executable.

   ```
   chmod +x ./setup.sh
   ```

6. Run the **setup.sh** script. Refer to the table below for additional arguments.

   ```
   ./setup.sh <additional-arguments>
   ```

7. Configure the service. Refer to the **Configure Hyland Preferences Service** section of this document.
8. Start the service.

   ```
   systemctl start hyland-preferences.service
   ```

Argument	Description	Default	Example
--install-dir, -i	This value is optional. It specifies the location to copy service files to when running setup.sh.	/opt/hyland-preferences-service	--install-dir /usr/shared/hyland-preferences-service

Argument	Description	Default	Example
--service-user, -u	This value is optional. It specifies the user the service should run as. If the user does not exist, they are created.	usrprefsvc	--service-user username

Configure Hyland Preferences Service

Configure Hyland Preferences Service with **environment variables**, or by editing the **appsettings.json** file in the installation directory. Any setting within the **appsettings.json** file can be set as an **environment variable** named in the format **HYLAND_PREFERENCES_SERVICE_<path-to-setting>**, where **<path-to-setting>** is the JSON path separated by two underscores. For example, setting the database provider would use the environment variable **HYLAND_PREFERENCES_SERVICE_Hyland.Preferences.Service.Database__Provider.**

Setting	Description
Host > WebServer > Http > Port	Specifies the port the service uses. This setting is ignored if the service is running in IIS. The default is 5600.
Host > WebServer > Https > Port	Specifies the port the service uses for secure requests. This setting is ignored if the service is running in IIS.
Host > WebServer > Https > Certificate > Path	Specifies the path to a .pfx (PKCS12) file used to serve as https.
Host > WebServer > Https > Certificate > Password	Specifies the password required to open the .pfx file.
Hyland.Logging > Routes > File > File	Specifies the log file path. The default is /logs/hsiPreferencesService-.txt.
Hyland.Logging > Routes > File > minimum-level	Specifies the log level. The default is Trace.
Hyland.Preferences.Service. Database > Provider	Specifies which data provider to use when connecting to the database. Valid options are SqlServer, PostgreSQL, and Oracle. The default is SqlServer.
Hyland.Preferences.Service. Database > ConnectionString	Specifies the connection string used to connect to the database. Oracle connection strings are required to use the Direct option set to true. Sample connection strings: **SqlServer** Server=localhost;Database=UserPreferences;User Id=usrprefsvc;Password=imagenow **Oracle** Server=localhost:1521/SID;User Id=usrprefsvc;Password=imagenow;Direct=true
User.Auth > SignedToken > Enabled	Specifies if authentication with signed tokens is enabled.
User.Auth > SignedToken > KeyIds	JWT **kid** header based configuration. This is an object that maps the **kid** header on a received JWT to the configuration used to validate the JWT signature.
User.Auth > SignedToken > KeyIds > [kid] > HashAlgorithm	Hash algorithm used to create the private/public key pair used to sign and validate the JWT. Valid values include RS256, RS384, RS512, ES256, ES384, ES512.

Setting	Description
User.Auth > SignedToken > KeyIds > [kid] > PublicKeyFile	Path to public key file used to validate JWT. Supports reading X509 certificates, PKCS8 encoded keys and PKCS7 objects.
User.Auth > SignedToken > KeyIds > [kid] > Scopes	Specifies the required scopes used during token validation. Separate multiple scopes with a single space.
User.Auth > SignedToken > KeyIds > [kid] > ClaimMapping > User	Name of the claim to use for user ID.
User.Auth > SignedToken > KeyIds > [kid] > ClaimMapping > System	Name of the claim to use for system. System and user make a user unique.
Note The following IDP User.Auth settings are optional.	
User.Auth > Identity > Enabled	Specifies if Identity based authentication is enabled.
User.Auth > Identity > Provider	Automatically configure ClientId, ClientSecret, and ClaimMapping for a known Identity Provider. Valid options are None and HylandIdp. The default is None.
User.Auth > Identity > ClientId	Client ID used during introspection if a reference token is used.
User.Auth > Identity > ClientSecret	Client secret used during introspection if a reference token is used.
User.Auth > Identity > Scopes	Required scopes, used during token validation. Multiple scopes can be separated with a single space.
User.Auth > Identity > RequireHttpsMetadata	Requires that the Authority uses HTTPS. Should always be true for production environments. Defaults to true.
User.Auth > Identity > ClaimMapping > User	Name of the claim to use for user ID.
User.Auth > Identity > ClaimMapping > System	Name of the claim to use for system. System and user make a user unique.
User.Auth > Identity > ValidTokenTypes	Specifies the token type. Multiple tokens can be separated with a comma. The default is “jwt”, “at+jwt”.

Example configuration

```
{
  "Host": {
    "WebServer": {
      "Http": {
        "Port": 5600
      },
      "Https": {
        "Port": 5643,
        "Certificate": {
          "Path": "/certificates/webserver.pfx",
          "Password": "password"
        }
      }
    }
  },
```

```
  "Hyland.Logging": {
    "Profile": "Default",
    "Routes": {
      "File": {
        "File": "d:/logs/hsiPreferencesService-.txt",
        "minimum-level": "Trace"
      }
    }
  },
  "Hyland.Preferences.Service.Database": {
    "Provider": "SqlServer",
    "ConnectionString": "Server=localhost;Database=UserPreferences;User
Id=usrprefsvc;Password=imagenow"
  },
  "User.Auth": {
    "Identity": {
      "Enabled": true,
      "Authority": "https://sample-idp.net/identityprovider",
      "ClientId": "introspection-client-id",
      "ClientSecret": "introspection-secret",
      "Scopes”: "user.profile psw.preferences-service",
      "ClaimMapping": {
        "User": "sub",
        "System": "tenant"
      },
      "ValidTokenTypes": ["type1", "type2"]
   }
  },
  "SignedToken": {
     "Enabled": true,
     "KeyIds": {
        "iTqXXI": {
          "HashAlgorithm": "RS256",
          "PublicKeyFile": "/certificates/preferences-service.pub",
          "ClaimMapping": {
            "User": "uid",
            "System": "sys"
          }
        }
      }
    }
  }
}
```

(Optional) Encrypt Hyland Preference Service Configuration

Hyland Preferences Service relies on another utility, **Hyland Application Settings Utility**, to encrypt and decrypt the **appSettings.json** file. The only setting currently configured to handle encryption is the **Hyland.Preferences.Service.Database:ConnectionString** setting. Follow the documentation provided for the Hyland Application Settings Utility to add encryption to your appSettings.json.

Note If you are installing on Linux, you must run the service with the same user that you registered certificates with Hyland Application Settings Utility. You can change this user in the systemmd service configuration file **/etc/systemd/system/hyland-preferences.service** or you can set it during the installation process by passing the **–u** option when running **setup.sh**.

Download the Hyland Application Settings Utility

To obtain Hyland product installation files and documentation, contact the Hyland Software Technical Support group. For a list of Technical Support phone numbers, go to hyland.com/pswtscontact.

About Hyland Preferences Service Authentication

You can authenticate Hyland Preferences Service requests using either a manually signed JWT or a token issued by an Identity Provider.

Signed token authentication uses a token created from a trusted source signed by an RSA private key, ECDSA private key or a shared secret. The service validates the token using a provided public key or the same shared secret the token was signed with.

You can use identity authentication with Identity Providers that support OAuth 2.0 token introspection. The provided bearer token is validated against the configured authority.

Configure Integration Server to use Hyland Preferences Service

Integration Server uses the Hyland Preferences Service to manage user preferences that are set through the preferences endpoint. The settings in the following table are defined in the **[Preferences]** group of the integrationserver.ini configuration file.

Note Restart the webserver that is hosting Integration Server after making any changes.

Setting	Options	Description
preferences.service.url	URL	The URL of the Preferences Service. The port used in the URL should match the Preferences Service configuration.
preferences.service.jwt.key.id	Any valid string. Recommend at least 6 characters.	The JWT kid header passed to the Preferences Service. This value should match a configuration in the Preferences Service **User.Auth > SignedToken > KeyIds** configuration
preferences.service.jwt.hash.algorithm	RS256 RS384 RS512 ES256 ES384 ES512	The JWT **alg** header passed to the Preferences Service. This configuration should match the configuration used in the Preferences Service.
preferences.service.keystore.location	Any valid file path location	The location of the Java KeyStore that contains the private key all JWT's are signed with. **Note** Use escaped backslashes “\\” or a forward slash “/” for the path separator.
preferences.service.keystore.password	Any valid string	The password to open the Java KeyStore
preferences.service.key.alias	Any valid string	The alias of the private key stored in the Java Keystore
preferences.service.key.password	Any valid string	The password required to access the private key
preferences.service.max.requests	Any positive number	The maximum amount of concurrent requests that can be made to the Preferences Service. The default is 8.
preferences.service.connectionpool.enabled	TRUE FALSE	Enables or disables connection pooling for Preferences Service requests

preferences.service.connectionpool.connection.cleanup.interval.seconds	Any positive number	Controls the cleanup interval for expired and idle Preferences Service pooled connections
preferences.service.connectionpool.connection.cleanup.after.idle.seconds	Any positive number	Controls the period of time that idle connections can remain in the Preferences Service connection pool

Example configuration

```
preferences.service.url=http://localhost:5600/
preferences.service.jwt.key.id=iTqXXI
preferences.service.jwt.hash.algorithm=RS256
preferences.service.keystore.location=D:\\dev\\certs\\prefs.ks
preferences.service.keystore.password=password
preferences.service.key.alias=keyalias
preferences.service.key.password=password
preferences.service.max.requests=8
preferences.service.connectionpool.enabled=TRUE
preferences.service.connectionpool.connection.cleanup.interval.seconds=3
preferences.service.connectionpool.connection.cleanup.after.idle.seconds=25
```

Private and public key creation

The Hyland Preferences Service uses asymmetric key validation of JSON web tokens (JWT), where a private key is used to sign the JWT during creation in Integration Server, and a public key is used to validate the signed JWT when received by the Hyland Preferences Service.

You can generate a key pair using your organizations best practices and import the key into a Java KeyStore that Integration Server has access to or generate a new keystore and key pair using Java's keytool utility. The key must use either the RSA or EC algorithm with a length of no less than 1024.

Note The service returns unauthorized errors if configured incorrectly. Refer to the service logs to determine any issues with setup.

To generate a new Java KeyStore and RSA key pair, complete the following steps.

1. Open a command window and navigate to the bin directory of your JRE installation.
2. To generate a new KeyStore and key pair, run the following command.

```
keytool -genkeypair -alias <alias> -keyalg RSA -keystore <path-to-keystore> -
keysize <keysize>
```

Example

```
keytool -genkeypair -alias prefs -keyalg RSA -keystore c:\cert\.keystore -keysize
2048
```

Setting	Options	Description
-alias	Any valid string.	Specifies the identifier associated with the key pair to be generated in the created KeyStore.
-keyalg	RSA	Specifies the algorithm used to generate the key pair.
-keystore	Any valid path.	Specifies the path where the KeyStore is created. **Note** Only JKS KeyStores are supported.
-keysize	Any valid number, minimum 1024.	Specifies the length of the key to be generated.

3. To export the certificate from the KeyStore, run the following command.

```
keytool –exportcert –keystore <path-to-keystore> -alias <alias> -rfc –file <path-to-certificate>
```

Example

```
keytool -exportcert -keystore c:\cert\.keystore -alias prefs -rfc -file c:\cert\prefcert.crt
```

Setting	Options	Description
-keystore	Any valid path.	Specifies the KeyStore path created in the previous step.
-alias	Any valid string.	Specifies the alias used when creating the key pair in the previous step.
-file	Any valid path.	Specifies the path where the certificate will be exported.

4. To export the the public key from the certificate, run the following command.

```
openssl x509 –pubkey –noout -in <path-to-certificate> > <path-to-public-key>
```

```
openssl x509 -pubkey -noout -in c:\cert\prefcert.crt > c:\cert\prefcert.pub
```

Setting	Options	Description
-in	Any valid path.	Specifies the certificate path exported in the previous step.
<path-to-public-key>	Any valid path.	Specifies the path where the public key will be exported.

5. Use the created KeyStore when configuring Intgeration Server and the exported public key when configuring the Hyland Preferences Service.

Appendix A: Troubleshooting

If you encounter any issues with the preferences service, refer to both the Integration Server logs and the Hyland Preferences Service logs to help aid in resolving the problem.

Healthcheck endpoint

If the service is running, you can check the health status of the service by making a request to **http://<server-address>:<port>/healthcheck**. This endpoint checks the general health of the service as well as the service's ability to connect to the configured database. The response body from the server is either **Healthy** or **Unhealthy**.

Authorization failed

If the Perceptive Content user is valid, **authorization failed** messages typically are a result of an invalid Java KeyStore configuration in the **integrationserver.ini**.

For example, the configured KeyStore path may not be accessible by Integration Server, or may not exist. Alternatively, check all passwords used to access the KeyStore, as well as the alias of the key in use.

Note Refer to the Integration Server logs for detailed exceptions.

Preferences Service starts but can't be connected to

First, check that the **integrationserver.ini** settings are using the correct URL to hit the preferences service. If that setting is accurate, and no logs are created by the preferences service, check that the **appsettings.json** is valid JSON, and that any configuration value that is restricted to a list of options uses a valid value.

401 unauthorized

Several situations can occur that result in **401 Unauthorized** errors, most commonly a result of incorrect **User.Auth** configuration in **appsettings.json**, or incorrect JWT configuration in **integrationserver.ini**. Assuming the Perceptive Content user credentials you are using are correct, check the following configurations.

- A configuration in **appsettings.json** exists for the *preferences.service.jwt.key.id* setting found in **integrationserver.ini**. The configuration can be found under **User.Auth > SignedToken > KeyIds.**
- The hash algorithm configured in the **integrationserver.ini** *preferences.service.jwt.hash.algorithm* setting, matches the hash algorithm configured in **appsettings.json.** The configuration in **appsettings.json** is found under **User.Auth > SignedToken > KeyIds > <keyId>**, where **<keyId>** is the *preferences.service.jwt.key.id* value configured in **integrationserver.ini**.
- Check that the public key file configured in **appsettings.json** exists and is in a supported format.

Note Refer to the Hyland Preferences Service logs for detailed exceptions.