Hyland DataTransfer

Installation and Setup Guide

Version: 7.0.x

Written by: Documentation Team, R&D
Date: July 2025

Hyland

Documentation Notice

Table of Contents

Prerequisites

Hyland DataTransfer is a data matching, review, and upload solution that allows you to automatically verify, clean up, and enter data extracted from hand-printed paper forms, downloadable web forms, and almost any other electronic source.

You can quickly and accurately upload information into your database, such as online and paper applications, recruit cards, SAT scores, ACT scores, GRE scores, and AP scores without the risk of creating duplicate student records.

With Hyland DataTransfer, you can complete the following tasks.

- Import a nearly unlimited range of data files.
- Match input data to records in your database.
- Review potentially matching records side-by-side.
- Upload new or modified information in real-time into your database.

Refer to the following topics before you install Hyland DataTransfer 7.0.x.

- Security Recommendations
- Network Security Recommendations
- Install the JDK
- Install Apache Tomcat
- Download Hyland DataTransfer

Security Recommendations

The following recommendations are for security.

- **End user authentication**
  Only system administrators have access to Hyland DataTransfer application. Regular end-users are not allowed to access the application.
- **Restricted Access**
  Hyland DataTransfer Web Application is accessible only as a localhost or comparable (from the same private network it is installed on along with the production database). This is to restrict access only to Administrator with access to this private network. No access outside of the private network is allowed.
- **Database user-account restrictions**
  Connection from Hyland DataTransfer backend to production Database must use a dedicated Database user-account. The account must be restricted to use minimum necessary tables required.
- **Dedicated Service to use Database**
  Only the dedicated admin-facing service (used by Administrator) can access the SQL queries table using its dedicated Database user account.

- **Private Network UI Restriction**
  The System Administrator must ensure that application's UI is only accessible on private network to prevent external attackers from reaching the vulnerable screen.

Network Security Recommendations

The following best practice recommendations are for network security.

- The Hyland DataTransfer server should be located behind a firewall, behind the enterprise intranet or internet.
- The Apache Tomcat instance that hosts Hyland DataTransfer 7.0.x should have SSL communication enabled and should be configured to use certificates from a known trusted signing authority.

  Note: For further securing your environment, refer to the Apache Tomcat Configuration Reference documentation.
- The database server should be located behind a firewall.

Install the JDK

Hyland DataTransfer 7.0.x requires the Java Development Kit (JDK). It is recommended that you use the JDK version 8.0. You can download the JDK from https://www.oracle.com/technetwork/java/javase/downloads/index.html.

You should install the JDK using the default installation settings.

Install Apache Tomcat

Note: Install the JDK before installing Apache Tomcat because the JDK is required for Tomcat installation.

Hyland DataTransfer requires Apache Tomcat. It is recommended that you use Tomcat, version 9.0.71 and JDK version 8.0.

You can download Apache Tomcat from the following websites.

- Apache Tomcat 9 Downloads

To enable SSL communication on the Tomcat server, refer to the information provided at the following websites.

- Apache Tomcat 9.0

If you are deploying Hyland DataTransfer in a Microsoft Windows environment, download and install the Windows Service Installer version.

After you install the Tomcat server, you must increase the default memory values. You can specify the memory values by configuring a system-wide environment variable, such as JAVA_OPTS as shown in the following example.

```
JAVA_OPTS="-Xms512m  -Xmx8192m"
```

Note: You should increase the Xmx value based on the available memory on the server. Optionally, you can use `-XX:MaxMetaspaceSize=<Metaspace Size>(m|g)`

For example, `-XX:MaxMetaspaceSize=1024m` to set an upper bound on the MetaSpace. This field is optional because this native memory region grows automatically by default. For more information, refer to the Oracle documentation.

To configure memory in Windows, complete the following steps.

1. Locate the **Apache Tomcat <version> Properties** dialog box using either of the following methods:
   - Click **Start**, choose **Programs**, select **Apache Tomcat <version>** and select **Configure Tomcat**. The **Apache Tomcat <version> Properties** dialog box displays.
   - Locate the tomcat<version> w.exe within the 'bin' directory of your Tomcat installation. For example, CATALINA_HOME/bin.
2. Click the **Java** tab.
3. In the **Maximum memory pool** field, type `8192`.
4. Click **Apply**.
5. To restart Tomcat, click the **General** tab, click **Stop**, and then click **Start**.

In Linux, run export `JAVA_OPTS="-Xms512m -Xmx8192m"` before you start Tomcat.

Download Hyland DataTransfer

To obtain the Hyland DataTransfer installation files, contact the Hyland Software Technical Support group.

Install and configure Hyland DataTransfer

Refer to the following topics for information on installing and deploying Hyland DataTransfer.

- Deploy Hyland DataTransfer
- Configure the Database
- Configure Authentication

Deploy Hyland DataTransfer

To deploy Hyland DataTransfer, complete the following steps.

Note: *$CATALINA_HOME* environment variable must be defined in a Linux environment.

1. Make sure that the Hyland DataTransfer installation file is owned by the same user who is running the Apache Tomcat server.
2. Make sure that the Tomcat server is running.
3. If the **DataTransfer.war** file has a version in the file name (such as **DataTransfer_7_0_0.war**), rename the file to remove the underscored version number so that the name is **DataTransfer.war**.
4. Copy the **DataTransfer.war** file to the ***CATALINA_HOME*/webapps** directory, where *CATALINA_HOME* is the Tomcat installation directory.

   Note: The Tomcat server automatically installs Hyland DataTransfer.
5. Copy the **EXAMPLE-datatransfer.properties** file from either the ***CATALINA_HOME*/webapps/DataTransfer/WEB-INF/classes** directory or from the downloaded ZIP

file to the ***CATALINA_HOME*/lib directory**, where ***CATALINA_HOME*** is the Tomcat installation directory.

6. Rename the **EXAMPLE-datatransfer.properties** file to **datatransfer.properties**.
7. Edit the **datatransfer.properties** file to configure the database connection to the database that will host your Hyland DataTransfer tables and allow you to connect to your local database and to configure other important properties. See the Configure the Database section for more information.

Configure the Database

To configure the database to work with Hyland DataTransfer 7.0.x, complete the following steps.

1. Obtain the latest version of the appropriate Java Database Connectivity (JDBC) driver for your database. For more information, see the Obtain the JDBC Driver section.
2. Edit the **datatransfer.properties** file. For more information, see the Edit the datatransfer.properties section.
3. Start the Tomcat server.
4. In a browser, navigate to https://*SERVERHOSTNAME*/DataTransfer, where *SERVERHOSTNAME* is the hostname of the Hyland DataTransfer server, to verify that you can access the log in page.

Obtain the JDBC driver

You should use the latest version of the JDBC driver for the database engine that you access. Obtain the latest JDBC drivers for the appropriate database from the following locations.

- **Oracle**

  https://www.oracle.com/technetwork/database/features/jdbc/index-091264.html

- **Microsoft SQL Server**

  https://msdn.microsoft.com/en-us/library/mt484311(v=sql.110).aspx

- **MySQL**

  https://dev.mysql.com/downloads/connector/j/

After you obtain the appropriate version, place it in the **CATALINA_HOME/lib** folder.

Edit the datatransfer.properties file.

In the datatransfer.properties file, you can modify the variables that configure the primary Hyland DataTransfer database connection.

You enable variables by including them in the file and setting their values with =. You can disable a variable by removing it from the file, commenting out the line with #, or for some variables, setting the variable to *false*. You do not need to configure every setting.

Note: Trailing spaces invalidate entries; use caution when copying or pasting any values to make sure that there are no trailing spaces.

See the Edit database connections section for more information about editing database connections.

Edit database connections

You can perform the following tasks when modifying database connections.

- Specify the database in which Hyland DataTransfer tables are stored (only Oracle, Microsoft SQL Server, and MySQL databases can host Hyland DataTransfer tables).
- Specify a database connection for Hyland DataTransfer authentication. See the Configure Authentication section for more information.
- The URL for each database connection is dependent on your database and type.
- You can specify the schema for Hyland DataTransfer tables with the dt_schema parameter (for example, `dt_schema=datatransfer`). If you do not provide a schema, the default value `nolijxfr` is used.

  Note: If you have upgraded to Hyland DataTransfer 7.0.x from a previous version and are using the same datatransfer.properties file, you must add this parameter to the file.

You must configure a superuser with full rights to create objects in your configured schema, including connection rights and enough quota space (if applicable) to host several tables and data. If you are using Microsoft SQL Server, you must set the superuser's default schema to your configured schema.

Examples for each database connection are commented out in the default datatransfer.properties file.

Refer to these topics for more information.

- Edit Database Connections for Microsoft SQL Server
- Edit Database Connections for Oracle
- Edit Database Connections for MySQL

Edit Database Connections for Microsoft SQL Server

Below is an example of the Microsoft SQL Server code that is included in the datatransfer.properties file.

```
#dt_dataSourceName=test
#dt_schema=datatransfer
#dt_user=nolijxfr
#dt_password=nolijxfr
#dt_driverClass=com.microsoft.sqlserver.jdbc.SQLServerDriver
#dt_jdbcUrl=jdbc:sqlserver://localhost:1433;databaseName=nolijxfr
#dt_databasePlatform=org.hibernate.dialect.SQLServer2008Dialect
#dt_liquibase_url=jdbc:sqlserver://localhost:1433;databaseName=nolijxfr
#dt_timeout=60
#dt_queryTimeout=60
#dt_maxpoolsize=50
#dt_threadcount=20
```

To enable a connection, remove the # symbols from each line or add # symbols to any inactive dt_lines not included in the preceding example.

In this example, *localhost* is the network name of the database server; however, it can also be the IP address.

The number following the localhost value (1433) is the port number.

Note the following information.

- The default value for dt_schema is *nolijxfr*.

- The dt_timeout parameter defines the maximum idle time, in seconds, for a connection session before it times out and is recycled for a new connection session in the connection pool. This parameter is optional; the default value is 60 seconds.
- The dt_queryTimeout parameter defines the maximum time, in seconds, that elapses when an SQL query executes before JDBC times out the request. This parameter is optional; it uses the default query timeout value for the database or JDBC driver, which depends on your configuration.
- The dt_maxpoolsize parameter applies to all database connections configured for connection groups; the value should be set to match the database connection that requires the smallest pool.
- The dt_threadcount parameter defines the total threads that can be used to process jobs in batch mode. The minimum value is 1; the maximum value is 20. If you use a value less than 1 or greater than 20, or if you do not provide the parameter, the default value is 20. For example, for single-threaded batch processing, add the line dt_threadcount=1 to the datatransfer.properties file.
- Ensure that your Microsoft SQL Server database is configured to allow SQL Server type authentication, or the database refuses the connection.

Edit Database Connections for Oracle

The Oracle example included in the datatransfer.properties file is shown below.

```
#dt_dataSourceName=test
#dt_schema=datatransfer
#dt_user=nolijxfr
#dt_password=nolijxfr
#dt_driverClass=oracle.jdbc.driver.OracleDriver
#dt_jdbcUrl=jdbc:oracle:thin:@localhost:1521:orcl
#dt_databasePlatform=org.hibernate.dialect.Oracle10gDialect
#dt_liquibase_url=jdbc:oracle:thin:@localhost:1521:orcl
#dt_timeout=60
#dt_queryTimeout=60
#dt_maxpoolsize=50
#dt_threadcount=20
```

To enable a connection, remove the # symbols from each line or add # symbols to any inactive dt_lines not included in the preceding example.

In this example, *localhost* is the network name of the database server; however, it can also be the IP address.

The number following *localhost* is the port number (*1521*) and the last field is the Oracle System ID (SID).

Note the following information.

- The default value for dt_schema is *nolijxfr*.
- The dt_timeout parameter defines the maximum idle time, in seconds, for a connection session before it times out and is recycled for a new connection session in the connection pool. This parameter is optional; the default value is 60 seconds.
- The dt_queryTimeout parameter defines the maximum time, in seconds, that elapses when an SQL query executes before JDBC times out the request. This parameter is optional; it uses the default query timeout value for the database or the JDBC driver, which depends on your configuration.
- The dt_maxpoolsize parameter applies to all database connections configured for connection groups; the value should be set to match the database connection that requires the smallest pool size.

- The dt_threadcount parameter defines the total threads that can be used to process jobs in batch mode. The minimum value is 1; the maximum value is 20. If you use a value less than 1 or greater than 20, or if you do not provide the parameter, the default value is 20. For example, for single-threaded batch processing, add the line dt_threadcount=1 to the datatransfer.properties file.

Edit Database Connections for MySQL

The MySQL example included in the datatransfer.properties file is shown below.

```
#dt_dataSourceName=nolijxfr
#dt_schema=nolijxfr
#dt_user=nolijxfr
#dt_password=nolijxfr
#dt_driverClass=com.mysql.jdbc.Driver
#dt_jdbcUrl=jdbc:mysql://localhost:3306/nolijxfr
#dt_databasePlatform=org.hibernate.dialect.MySQL5Dialect
#dt_liquibase_url=jdbc:mysql://localhost:3306/nolijxfr
#dt_timeout=60
#dt_queryTimeout=60
#dt_maxpoolsize=50
#dt_threadcount=20
```

To enable a connection, remove the # symbols from each line and remove or add # symbols to any active dt_lines not included in the preceding example.

In this example, *localhost* is the network name of the database server; however, it can also be the IP address.

The number following *localhost* is the port number (*3306*) and the last field is the database name.

Note the following information.

- The default value for dt_schema is *nolijxfr*.
- The dt_timeout parameter defines the maximum idle time, in seconds, for a connection session before it times out and is recycled for a new connection session in the connection pool. This parameter is optional; the default value is 60 seconds.
- The dt_queryTimeout parameter defines the maximum time, in seconds, that elapses when an SQL query executes before JDBC times out the request. This parameter is optional; it uses the default query timeout value for the database or JDBC driver, which depends on your configuration.
- The dt_maxpoolsize parameter applies to all database connections configured for connection groups; the value should be set to match the database connection that requires the smallest pool size.
- The dt_threadcount parameter defines the total threads that can be used to process jobs in batch mode. The minimum value is 1; the maximum value is 20. If you use a value less than 1 or greater than 20, or if you do not provide the parameter, the default value is 20. For example, for single-threaded batch processing, add the line dt_threadcount=1 to the datatransfer.properties file.

Configure Authentication

Authentication in Hyland DataTransfer is configured in the datatransfer.properties file. The auth.type parameter in the datatransfer.properties file specifies which authentication is used. If this parameter is not supplied, Database Authentication is used.

Note: In Hyland DataTransfer 7.0.x, you do not need to configure the datatransfer-security.xml file manually.

Hyland DataTransfer 7.0.x supports datatransfer.properties file configuration for the following authentication techniques.

- Database authentication
- Centralized Authentication Service (CAS)
- Lightweight Directory Access Protocol (LDAP)
- Microsoft Active Directory

Each authentication type only determines access to Hyland DataTransfer. If a user has not been fully configured to use Hyland DataTransfer, the user is denied access to the application.

Refer to the following topics for more information.

- Configure Database Authentication
- Configure CAS Authentication
- Configure LDAP and Active Directory Authentication

Configure Database Authentication

Database authentication uses database user accounts to determine authentication to the application.

If you do not set authentication configuration in the datatransfer.properties file, Database Authentication is used by default; however, you can specify it explicitly with `auth.type=DB`.

In the datatransfer.properties file, use the following parameters.

- `auth_database.driver=oracle.jdbc.driver.OracleDriver`
- `auth_database.url=jdbc:oracle:thin:@oracle.Hyland.com:1521:sis`

Hyland DataTransfer supports a wide variety of database types as authentication providers. A JDBC driver must be present in the CATALINA_HOME/lib directory. After you copy the driver into the directory, you must restart Tomcat before the driver is available for use. See the Obtain the JDBC Driver section for more information.

Configure CAS Authentication

If a user is using CAS, the user name and password entered by the user at the Hyland DataTransfer login page can be authenticated against the CAS server.

Note: Add the certificate used in the CAS server to the cacerts of JRE used by Tomcat.

When you use CAS as an authentication method, you must complete the following tasks.

- Every time you deploy a new version of Hyland DataTransfer, rename the datatransfer-security.xml to another filename, such as datatransfer-security.xml.original.
- Next, rename the datatransfer-security.xml.cas.xml file to datatransfer-security.xml and restart the server.
- Configure the datatransfer.properties file by setting the following parameters.

  Note: `auth.CasNtUrl` has a forward slash (/) at the end.

  - `auth.type=CAS`. Configures CAS as the protocol.

- `auth.CasNtUrl=https://my.pdtserver.org/DataTransfer/` Specifies the Hyland DataTransfer URL.
- `auth.CasLoginUrl=https://cas.example.org/cas/login` Specifies the CAS login URL on the CAS server.
- `auth.CasServerUrl=https://cas.example.org/cas` Specifies the CAS server URL.
- `auth.CasSendRenew=false`
- `auth.CasLogoutUrl=https://cas.example.org/cas/logout`

Configure LDAP and Active Directory Authentication

Refer to the following information for more information.

- LDAP and Active Directory concepts
- Configure Hyland DataTransfer against LDAP and Active Directory
- Configure LDAP and Active Directory

LDAP and Active Directory concepts

For information on understanding LDAP and Active Directory Concepts, see the following information.

- Understanding Binding
- Understanding Searching
- Migrate from Database Authentication to Directory Services

Understanding Binding

Binding is the act of submitting credentials (typically, the user name and password) to the directory service and receiving a binary ‘yes’ or ‘no’ authentication. After a credential is bound, other properties of the credential, such as phone number or role, can be searched.

Typically, directory services allow binding only using a Common Name (CN) attribute and a password. Because directory services are flexible, the CN can be anything the local administrator wants. Usually, the CN is either a user's full name or user name.

Understanding Searching

Some enterprises have portions of their directories publicly searchable, so that you can search for a CN if you have another attribute. Others require authentication before a search can be completed and have generic accounts established for this purpose.

Migrate from Database Authentication to Directory Services

The directory service must contain an attribute that matches Hyland DataTransfer user name (USERVAL_USER_ID).

Configure Hyland DataTransfer against LDAP and Active Directory

Before you configure LDAP, you should understand that LDAP topology and requirements of the user. Consider the following guidelines.

- Whether the LDAP connection is secure.
- The LDAP URL.
- Attributes, such as full name, user name, or employee ID, to be used when logging into Hyland DataTransfer.
- Whether the directory service supports anonymous queries.
- Whether the directory service contains a value that matches Hyland DataTransfer user name (USERVAL_USER_ID).
- Information required for Hyland DataTransfer to query in the LDAP structure to find user IDs.
  - The attribute name, uid, and sAMAccountName of the directory service.
  - The group to which Hyland DataTransfer users belong and the appropriate query string for the group.

Configure LDAP and Active Directory

In the datatransfer.properties file, configure the following parameters.

- `auth.type=LDAP` Configures LDAP as the authentication protocol.
- `auth.LdapConnString=ldap://ldap.myuniversity.edu:389/dc=myuniversity,dc=edu` Configures the connection string for the LDAP server.
- `auth.LdapBrowseUserDn=ldapbrowse@myuniversity.edu` Specifies a user who has browse permissions on the LDAP server.

  Note: Any DN that can be interpreted by the LDAP Server is acceptable.
- `auth.LdapBrowsePassword= ldappassword` Specifies the password for the user with browse permissions on the LDAP server.
- `auth.LdapOu=OU=SBSUsers,OU=Users,OU=MyBusiness` Specifies the Organizational Unit (OU) used to find users.

  Note: This performs a subtree search.
- `auth.LdapFilter=(CN={0})` Optional parameter as the default value is `SAMAccountName={0}`. Specifies the filter to match the entered user name.
- `auth.LdapMemberOf=CN=HylandUsers,OU=Security Groups,DC=myuniversity,DC=edu` Optional. Specifies a CN group to limit users.

  Note: A subtree search is not performed.

Licensing Hyland DataTransfer

Refer to the following section for information on licensing Hyland DataTransfer.

Obtain and Install a License

You require a license to operate a Hyland DataTransfer server. Without a license, you are unable to log in to Hyland DataTransfer or run scheduled jobs. The license is contained in a file named *datatransfer.lic* and is not usable with other Hyland products. One license is required for each computer running a Hyland DataTransfer server, and it is IP address-restricted. After you have installed a license, if the IP address of the computer changes, it will invalidate the license, and you will need to obtain a new one.

Your license may be provided by a Hyland consultant who is installing or upgrading your Hyland DataTransfer instance; otherwise, you must contact Hyland Software Technical Support and request a license.

To install a license, copy the datatransfer.lic file to your *CATALINA_HOME*/lib directory. You must restart Tomcat before the license is activated.

Upgrade to Hyland DataTransfer 7.0.x

This section describes the process for upgrading to Hyland DataTransfer 7.0.x in a TEST environment. This process, or a subset of it, must also be applied to a PROD (production) environment. Before you upgrade Hyland DataTransfer, you should place a new and complete copy of PROD in a dedicated TEST environment.

Refer to *Hyland DataTransfer System Overview* for information about client and server hardware and software requirements for Hyland DataTransfer 7.0.x.

Updates to Hyland DataTransfer tables are automatically completed when the new version of Hyland DataTransfer is installed and running in Tomcat.

To upgrade to Hyland DataTransfer 7.0.x, complete the following steps.

1. Obtain the latest **DataTransfer.war** file. See the Download Hyland DataTransfer section for more information.
2. Deploy Hyland DataTransfer. See the Deploy Hyland DataTransfer section for more information.

   Note: You must remove any pre-existing Hyland DataTransfer folder before deploying the DataTransfer.war file.
3. If you are upgrading from a version earlier than 7.0 x and are using Microsoft SQL Server as your database for the Hyland DataTransfer tables, restart Tomcat after you install the new **DataTransfer.war** file.
4. If you are upgrading from Hyland DataTransfer 6.0.x or 6.1.x to Hyland DataTransfer 7.0.x, obtain a new license file (**datatransfer.lic**). You cannot rename the previous **nolijtransfer.lic** file to **datatransfer.lic**. See the Obtain and Install a License section for more information.

Beginning with Hyland DataTransfer 6.2.x, the property names in the datratransfer.properties file (previously named the nolijtransfer.properties file) have changed. Parameters that begin with *nt_* now start with *dt_*, which applies to the following parameters.

- `dt_dataSourceName`
- `dt_schema`
- `dt_user`
- `dt_password`
- `dt_driverClass`

- `dt_jdbcUrl`
- `dt_databasePlatform`
- `dt_liquibase_url`

The default value of the dt_schema parameter is *nolijxfr*.

You must change these parameters when upgrading to Hyland DataTransfer 7.0.x. See the Edit the datatransfer.properties file section for more information.