Oracle® Database

Using Oracle Data Transforms

F51654-28
June 2025

Oracle Database Using Oracle Data Transforms,

F51654-28

Primary Author: Oracle Corporation

Contents

Preface

This guide helps you to get started and work with Oracle Data Transforms. This includes creating the stack, provisioning the repositories, accessing the instance, creating connections, loading data, and running data flows to transform your data.

This preface contains the following topics:

Audience

This document is intended for administrators and users who want to deploy an Oracle Data Transforms instance and use it to load and transform data.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in Oracle Data Integrator Library:

- *Release Notes for Oracle Data Integrator*
- *Understanding Oracle Data Integrator*
- *Developing Integration Projects with Oracle Data Integrator*
- *Installing and Configuring Oracle Data Integrator*
- *Upgrading Oracle Data Integrator*
- *Integrating Big Data with Oracle Data Integrator Guide*
- *Application Adapters Guide for Oracle Data Integrator*
- *Developing Knowledge Modules with Oracle Data Integrator*

- *Connectivity and Knowledge Modules Guide for Oracle Data Integrator Developer's Guide*
- *Oracle Data Integrator Tools Reference*
- *Data Services Java API Reference for Oracle Data Integrator*
- *Open Tools Java API Reference for Oracle Data Integrator*
- *Getting Started with SAP ABAP BW Adapter for Oracle Data Integrator*
- *Java API Reference for Oracle Data Integrator*
- *Getting Started with SAP ABAP ERP Adapter for Oracle Data Integrator*
- *Oracle Data Integrator 14c Online Help*, which is available in ODI Studio through the Help Center when you press **F1** or from the main menu by selecting **Help**, and then **Search** or **Table of Contents**.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1
Overview

Learn about Oracle Data Transforms and its concepts.

Topics

- Introduction to Oracle Data Transforms
  Oracle Data Transforms allows you to design graphical data transformations in the form of data loads, data flows, and workflows. Data Loads provide a convenient way of loading data into Oracle Autonomous Database.
- Supported Browsers
  Oracle Cloud Infrastructure supports the latest desktop versions of Google Chrome, Microsoft Edge, Internet Explorer 11, Safari, Firefox, and Firefox ESR.
- Supported Database Functions
  Oracle Data Transforms supports various database functions that you can drag and drop on the Design Canvas to connect components within a data flow.

1.1 Introduction to Oracle Data Transforms

Oracle Data Transforms allows you to design graphical data transformations in the form of data loads, data flows, and workflows. Data Loads provide a convenient way of loading data into Oracle Autonomous Database.

Oracle Data Transforms provides an easy-to-use user interface to create your data loads, data flows, and workflows without requiring you to write any code. Data flows define how the data is moved and transformed between different systems, while the workflows define the sequence in which the data flows are executed. You can drag and drop the components, that is data entities and operators, such as Join, Split, and Aggregate, on the Design Canvas of the Data Flow editor and connect them to complete you data flow. You can then add your data flows to workflows.

After your data flows and workflows are ready, you can execute them immediately or schedule to execute them at a later time. Oracle Data Transforms run-time agent orchestrates the execution of jobs. On execution, Oracle Data Transforms generates the code for you.

Oracle Data Transforms integrates with Oracle Autonomous Database (Autonomous Data Warehouse and Autonomous Transaction Processing) for repository creation as well as for loading data. The Oracle Data Transforms Repository is automatically setup at provisioning. Objects developed or configured by the users are stored in this repository.

Oracle Data Transforms has many concepts similar to Oracle Data Integrator (ODI), however, the terminology used is different. If you are familiar with ODI, then reviewing the Terminology Information section may be helpful to you. See Terminology Information.

You can launch Data Transforms in any of the following ways:

- **Oracle Cloud Marketplace:** Create a Data Transforms instance from Oracle Cloud Marketplace. Data Transforms is available as a separate listing on Marketplace called Data Integrator: Web Edition.

- **Autonomous Database Data Tools Page:** Navigate to Autonomous Database Data Tools Page, and click **Data Transforms** in the Database Actions page, or click the Selector icon and select **Data Transforms** from the Data Tools menu in the navigation pane.
  If you have already registered a Data Transforms instance from OCI Marketplace with the Autonomous Database, the **Data Transforms** card on the Database Actions page will continue to take you to your Marketplace instance

Access to the standard set of Data Transforms features may depend on where you launch Data Transforms from. In this documentation, certain topics could include any of the following badges to indicate content that may or may not be available for use:

- APPLIES TO: ● Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.
- APPLIES TO: ● Data Transforms instance that is registered with Autonomous Database.
- APPLIES TO: ● Data Transforms that is part of the suite of data tools built into Oracle Autonomous Database.

1.2 Supported Browsers

Oracle Cloud Infrastructure supports the latest desktop versions of Google Chrome, Microsoft Edge, Internet Explorer 11, Safari, Firefox, and Firefox ESR.

Note that mobile browsers as well as private browsing mode is not supported for Firefox, Internet Explorer, or Edge.

1.3 Supported Database Functions

Oracle Data Transforms supports various database functions that you can drag and drop on the Design Canvas to connect components within a data flow.

The Database Functions toolbar in the Data Flow editor includes the following database functions that can be used in your data flows. See Oracle Database SQL Language Reference for information about the database functions.

1. **Data Transformation**
   It contains the following components:
   - Aggregate
   - Expression
   - Filter
   - Join
   - Distinct
   - Lookup
   - Set
   - Sort
   - Subquery Filter
   - Table Function
2. **Data Preparation**
   It contains the following components:
   - Data Cleanse

- Substitution
- Equi_Width Binning
- Quantile Binning
- Lead
- Lag
- Replace

3. **Machine Learning**
   It contains the following components:
   - Prediction
   - Predition Model
   - Outlier Detection
   - Text Embedding Vector
4. **Text**
   It contains the following components:
   - REGEXP COUNT
   - REGEXP INSTR
   - REGEXP SUBSTR
   - REGEXP REPLACE
   - Edit Distance Similarity
   - Contains
5. **Oracle Spatial and Graph**
   It contains the following components:
   - Buffer Dim
   - Buffer Tol
   - Distance Dim
   - Distance Tol
   - Nearest
   - Simplify
   - Point
   - Geocode Tools:

     > **Note:**
     >
     > The following Geocode Tools work only in non-Autonomous Database environment.

     - Geocode As Geometry
     - Geocode
     - Geocode Addr
     - Geocode All

  - Geocode Addr All
  - Reverse Geocode

> **Note:**
>
> The following Geocode Tool works only in an Autonomous Database environment.

  - Geocode Cloud
- Spatial Join

2
Prerequisites

Make sure you have the following prerequisites before using Data Transforms on Oracle Cloud:

- Oracle Cloud Account
- Have access to assigned Oracle Cloud Tenant
- Compute node resources within Oracle Cloud Tenant

Go through the following prerequisites carefully before creating the Oracle Data Transforms instance:

- Refer to Create Dynamic Group and Policies for compartment, before provisioning Oracle Data Transforms and Autonomous Database Instances.
- Read the password policies mentioned in Manage Password Complexity on Autonomous Database. The password that you provide should conform to the password validation rules of the CLOUD_VERIFY_FUNCTION, which is the default password verification function for Autonomous Database.

Topics

- Create an SSH/RSA Key
  To work with the Oracle Cloud Infrastructure once the Oracle Data Transforms Compute Node is built, you have to provide a SSH Public Key to allow you to login to the node.
- Create Dynamic Group and Policies
  Dynamic Group and Policies are required when using the Oracle Data Transforms Repository on an Autonomous Database.

2.1 Create an SSH/RSA Key

To work with the Oracle Cloud Infrastructure once the Oracle Data Transforms Compute Node is built, you have to provide a SSH Public Key to allow you to login to the node.

In order to build your SSH keys, perform the following steps:

1. In a terminal window, generate the SSH key using the following command:

   ```
   $ ssh-keygen Generating public/private rsa key pair.
   ```

2. Enter the path to store this file. By default, this gets saved in your home directory under a hidden folder called .ssh. Change this default location, if required.

   ```
   Enter file in which to save the key (/Users/johndoe/.ssh/id_rsa): <Return>
   ```

3. Enter a passphrase using your key.

   ```
   Enter passphrase (empty for no passphrase): <passphrase>
   ```

4. Re-enter the passphrase to confirm it.

```
Enter same passphrase again: <passphrase>
```

5. Check the results.
The key fingerprint (a colon separated series of two digit hexadecimal values) is displayed. Check if the path to the key is correct. In the above example, the path is `/Users/johndoe/.ssh/id_rsa.pub`. You have now created a public and private key pair.

2.2 Create Dynamic Group and Policies

Dynamic Group and Policies are required when using the Oracle Data Transforms Repository on an Autonomous Database.

During instance provisioning, data servers for all accessible Autonomous Databases (ADB) are automatically created, as long as the dynamic group and policies are created before deploying Oracle Data Transforms on Marketplace and are set as mentioned below.

The provisioning wizard provides two options for you to select the autonomous database where the compute node will be built in. You need to create the dynamic group and policies depending on the option you want to use.

- Search for the autonomous database using the OCID -
To use this option create a dynamic group to include the OCID of the compartment that contains the database and the instance. For example,

```
ALL {instance.compartment.id =
'ocid1.compartment.oc1..aaaaaaaabgr34tpuanpvq6xfb667nsmy2jz45zj6dexojhxdsv4
mjayem3cq'}
```

You have to setup the following policy at the compartment level:

```
Allow dynamic-group <group-name> to inspect autonomous-database-family in
compartment <compartment-name>
Allow dynamic-group <group-name> to read autonomous-database-family in
compartment <compartment-name>
Allow dynamic-group <group-name> to inspect compartments in compartment
<compartment-name>
```

For tenancies that use identity domains, you need to include the domain name along with the group name when you setup the policy. For example:

```
Allow dynamic-group <identity-domain-name>/<group-name> to inspect
autonomous-database-family in compartment <compartment-name>
Allow dynamic-group <identity-domain-name>/<group-name> to read autonomous-
database-family in compartment <compartment-name>
Allow dynamic-group <identity-domain-name>/<group-name> to inspect
compartments in compartment <compartment-name>
```

If the autonomous database and Data Transforms instance are in different subcompartments you need to create a dynamic group to include the OCID of the compartment that contains the instance. You then need to define a policy that grants the dynamic group access to the compartment that contains the database.

For example, let's assume

- RootCompartment
- SubcompartmentA - contains the autonomous database
- SubcompartmentB - contains the Data Transforms instance

In this case, if you want to search and provision using the OCID of SubcompartmentA you need to do the following:

- Create a dynamic group, say `instance_grp`, to include the OCID of SubcompartmentB. For example,

```
ALL {instance.compartment.id = 'OCID of SubcompartmentB'}
```

You then have to setup the following policy to allow `instance_grp` access to SubcompartmentA:

```
Allow dynamic-group instance_grp to inspect autonomous-database-family
in compartment SubcompartmentA
Allow dynamic-group instance_grp to read autonomous-database-family in
compartment SubcompartmentA
Allow dynamic-group instance_grp to manage instance-family in
compartment SubcompartmentA
```

- Browse to select the database -
To use this option create a dynamic group to include the compartment IDs of all the compartments within the tenancy (root compartment). For example,

```
ALL {instance.compartment.id =
'ocid1.compartment.oc1..aaaaaaaabgr34tpuanpvq6xfb667em3cnsmy2jz45zj6dexojhx
dsv4mjay4q'}
ALL {instance.compartment.id =
'ocid1.compartment.oc1..aaaaaaaa3elin667nsmpvq4fbytpua6x2jz4jayem35ojhxdsv4
mzj6dexcq'}
ALL {instance.compartment.id =
'ocid1.compartment.oc1..aaaaaaaa7g4jr3pvq6xfb4jz45tpuan667nmsyz42j5zj6dexoj
hxdsv4mja'}
```

You have to setup the following policy at the tenancy (root compartment) level:

```
Allow dynamic-group <group-name> to inspect autonomous-database-family in
compartment tenancy
Allow dynamic-group <group-name> to read autonomous-database-family in
compartment tenancy
Allow dynamic-group <group-name> to inspect compartments in compartment
tenancy
```

To configure email delivery service for specified groups on Oracle Cloud Marketplace:

An email approved sender must be in a group that has IAM policy permissions to send emails. An approved sender must be in a compartment with permissions to manage approved senders. You have to create a policy to manage approved senders in the entire tenant, if the approved senders exist in root compartment.

Add the following policy statement to enable `odi_group` to manage approved senders:

```
Allow dynamic-group odi_group to use approved-senders in compartment odi
```

3
Deploy

Know how to find Data Transforms on Oracle Cloud Marketplace, as well as steps to create the compute instance and provision the metadata repository.

Topics

- Select the Product
  Oracle Data Transforms is listed on the Oracle Cloud Marketplace as Data Integrator: Web Edition.
- Create the Oracle Data Transforms Instance
  This topic contains step-by-step instructions for creating the Oracle Data Transforms instance.
- Add Ingress Rules to the Default Security List
  For establishing communication with the instance, you need to ensure that the underlying Security List that is associated with the VCN has all the IP protocols enabled.
- Provision the Oracle Data Transforms Metadata Repository
  The very first screen you will see after you access the Oracle Data Transforms URL is the Administration tab, which provides easy wizards that allow you to either create a new repository in an Autonomous Database to store your transformations, or connect to an existing Data Transforms repository.

3.1 Select the Product

Oracle Data Transforms is listed on the Oracle Cloud Marketplace as Data Integrator: Web Edition.

To search for the product Oracle Data Transforms on Oracle Cloud Marketplace, enter “Data Integrator: Web Edition” as the product name in the search text box and click **Go**. From the displayed listings click **Data Integrator: Web Edition**. You are navigated to the product listing page which has all the basic usage information along with the product documentation.

> **Note:**
>
> You may also see a listing called Data Integrator: Web Edition (Versions prior to May 2022), which includes older versions of Data Integrator: Web Edition. Do not use this listing. Oracle recommends that you use Data Integrator: Web Edition to get the latest version.

1. From your respective product listing page, select **Get App**.
2. Select OCI Region or Login using your Single Sign-On credentials.
   - OCI Region – Select the desired region and click **Launch Image**.
3. Provide the OCI tenant details.
4. Sign-in to the Identity provider.

5. On the **Data Integrator: Web Edition** page, provide the following information:
   a. Type - It is Image by default.
   b. Version - Select a version from the drop-down. The latest version is selected by default.
   c. Compartment - Specifies the compartment where the compute node will be built. It is generally the location that you have access to build the compute node.
   d. Terms of Use - This check-box is selected by default. Oracle recommends to review the licenses before proceeding.
   e. Launch Instance - It launches the instance in the OCI environment. After selecting all the required information, click **Launch Instance**.

> **Note:**
>
> Provisioning your instance on Oracle Cloud Marketplace greatly depends on your product selection. You cannot use the same VM instance for multiple product offerings. The procedure listed below are product specific and may differ based on the product version.

3.2 Create the Oracle Data Transforms Instance

This topic contains step-by-step instructions for creating the Oracle Data Transforms instance.

Follow the below procedure to create your Oracle Data Transforms instance:
Once you click **Launch Instance**, you are navigated to the **Create compute instance** page.

1. Fill in the required information:
   - Name - Name of the Instance. This field displays a default name based on the timestamp. You can edit this field, if required.
   - Create In Compartment – This defaults to the compartment you have selected on the Oracle Data Integrator for Oracle page.
2. In the Placement settings, choose from any of the displayed availability domains. Do not change the values selected for **Capacity type** and **Fault Domain**.
3. In the Image and Shape settings, click **Change Shape** to select a template that you want to base the instance on.
4. In the Networking settings, fill in the required details to configure the networking options:
   - Primary network - Do one of the following:
     – Choose **Select exiting virtual cloud network**, and then select the desired VCN for the newly created instance from the **Virtual cloud network in *compartment*** drop-down. A VCN is a software-defined network that you set up in the Oracle Cloud Infrastructure data centers in a particular region. Click **Change Compartment** if you want to change the VCN compartment where the compute node will be built.
     – Choose **Create new virtual cloud network**, enter a name in **New virtual cloud network name** if you want to change the pre-populated value, and select the VCN compartment where the compute node will be built from the **Create in compartment** drop down.

You must add ingress rules to the default security list of the new created VCN to access the instance. See Add Ingress Rules to the Default Security List for more information.

- Choose **Enter subnet OCID** and enter the OCID of the compartment where the instance-related artifacts will be created in **Subnet OCID**.

- Subnet - Do one of the following
  - Choose **Select existing subnet** and then select a subnet from the **Subnet in *compartment*** drop-down. If you want to change the compartment use **Change Compartment** to choose a different compartment.

    > **Note:**
    >
    > You cannot use this option if you choose **Create new virtual cloud network** or **Enter subnet OCID** in the Primary network settings.

  - Select **Create new public subnet** and enter the following details:
    * New subnet name - Enter a name if you want to change the pre-populated value.
    * Create in compartment - If you want to change the compartment select a different compartment from the drop-down.
    * CIDR block - Specify the Virtual Cloud Network (VCN) CIDR block for the VCN if you want to change the pre-populated value. You cannot change this value later.

    > **Note:**
    >
    > You cannot see this option if you choose **Enter subnet OCID** in the Primary network settings.

- Public IP address - Do one of the following:
  - Assign a public IPv4 address - The newly created VM is assigned a public IP address.
  - Do not assign a public IPv4 address - No public IP address is assigned to prevent public access to the compute node.

5. Add SSH keys so that you can connect to the instance using SSH. You can choose from the following options:
   - Generate a key pair for me – Use this option to generate an SSH key pair to connect to the instance using an SSH connection.
   - Upload public key files (.pub) – Use this option if you want to upload a .pub file that contains the SSH key pair.
   - Paste public keys – Use this option if you want to manually specify the SSH key pair.
   - No SSH keys - Use this option if you do not want to enable SSH access.
6. Leave the boot volume selection as is.
7. Click **Create**.

8. When you get a message `Successful` from the OCI console, the instance is created. This takes approximately 5 minutes to complete.
9. After approximately 5 minutes, try to access the Data Transforms URL via a browser using the format `http://<public-ip address>:9999/oracle-data-transforms`. You can get IP address from the OCI Compute Instance properties page.

3.3 Add Ingress Rules to the Default Security List

For establishing communication with the instance, you need to ensure that the underlying Security List that is associated with the VCN has all the IP protocols enabled.

The following table lists the ingress rules you need to set to access your Data Transforms instance.

Table 3-1 Ingress Rules Table

Stateless	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows
No	0.0.0.0/0	TCP	All	22	Nil	TCP traffic for ports: 22 SSH Remote Login Protocol
No	10.0.0.0/24	TCP	All	80	Nil	TCP traffic for ports : 80
No	0.0.0.0/0	All protocols	Nil	Nil	Nil	All traffic for all ports
No	0.0.0.0/0	ICMP	Nil	Nil	3, 4	ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set

3.4 Provision the Oracle Data Transforms Metadata Repository

The very first screen you will see after you access the Oracle Data Transforms URL is the Administration tab, which provides easy wizards that allow you to either create a new repository in an Autonomous Database to store your transformations, or connect to an existing Data Transforms repository.

To provision the Oracle Data Transforms Repository on an Autonomous Database you need to create dynamic groups and policies depending on the option you want to use. Refer to Create Dynamic Group and Policies for details.

Follow the below procedure to provision the Oracle Data Transforms metadata repository:

1. On the Administration tab, select one of the following options:
   - Create a new Repository in an Autonomous Database
   - Connect to an existing Repository in an Autonomous Database

The repo mode created during configuration is marked as Oracle Data Transforms mode.

> **Note:**
>
> Oracle Data Integrator Repositories that you create during provisioning are mode specific. You cannot use a repository created in Oracle Data Transforms mode with a stack in ODI Studio mode and you cannot access a repository created in ODI Studio mode with a stack in Oracle Data Transforms mode.

2. In the Autonomous Database Details page, do one of the following:
   - Select **Use OCID** and then enter the OCID of the autonomous database that you want to use to search for the compartment where the compute node will be built in the database.
     To use this option create a dynamic group to include the OCID of the compartment where the database instance is created, and then set up a policy to grant access to the dynamic group at the compartment level.

     If the database and instance are in different subcompartments you need to create a dynamic group to include the OCID of the compartment that contains the instance. You then need to define a policy that grants the dynamic group access to the compartment that contains the database.
   - Select **Browse for Database** and then do the following:
     a. Autonomous Database Compartment - Select the compartment where the compute node will be built in the database.
     b. Autonomous Database Instance - Select the created Autonomous Database instance from this drop down arrow.

     To use this option create a dynamic group to include the compartment IDs of all the compartments within the tenancy (root compartment) and then set up a policy to grant access to the dynamic group at the tenancy (root compartment) level.

   Refer to Create Dynamic Group and Policies for details about creating dynamic groups and policies for each option.
3. When you select the required Autonomous Database instance, Data Transforms generates a default password to download the wallet to create a connection to that instance. If you need to specify a custom password that conforms to specific validation rules, select the **Use custom wallet password** check box, and enter the password in the **Wallet Password** field.
4. Click **Fetch Service Names** to retrieve the services associated with the database and select a name from the drop-down. Make sure that you select the service name with the option `_low` from the drop-down.
5. Provide the password credentials of the administrator user of the Autonomous Database in the **Autonomous Database Admin Password** textbox.
   The password that you provide should conform to the password validation rules of the CLOUD_VERIFY_FUNCTION, which is the default password verification function for Autonomous Database. For more information see, Manage Password Complexity on Autonomous Database.
6. In the Schema Details page, enter the following:
   - **Schema Prefix** - Specify the prefix used for the Oracle Data Transforms Schemas to identify them.

> **Note:**
>
> A valid prefix should contain only alpha-numeric characters. It should not start with a number and should not contain any special characters. A maximum of 12 characters are allowed.

For existing repository, the Schema Prefix field is a drop down that lists the schema prefixes available based on the Autonomous Database Instance you selected in the Autonomous Database Details page.

- **Schema Password** - Specify the password used for accessing the Oracle Data Transforms schema.

> **Note:**
>
> The password that you provide should conform to the password validation rules of the CLOUD_VERIFY_FUNCTION, which is the default password verification function for Autonomous Database. For more information see, Manage Password Complexity on Autonomous Database.

For existing repository, provide the password configured during repository creation. Note that the schema password expires in 180 days from the date when the repository was created. In such cases, provisioning fails when you try to connect to the existing repository. See Reset the Schema User Password for instructions on how to reset the schema password.

7. In the Repository Details page, provide the ODI Supervisor password.

> **Note:**
>
> The Supervisor password that you provide should:
>
> - contain alpha-numeric characters,
> - be 6-12 characters in length,
> - not start with a number,
> - not contain any special characters except $, # and _.

8. Click **Save**. The final screen of the wizard displays all the processes that run to complete the configuration. This takes about 15 minutes. The screen auto-refreshes to display the status of each process. You can also click the Refresh icon to view the status of the configuration.
9. Click **Download Log File** to download the configuration logs. The log files are downloaded in a zip file named `provlogs.zip`.
10. When the configuration completes successfully, the **Login to Data Transforms** button is activated, which allows you to navigate to the Oracle Data Transforms login page.

If the configuration fails at any step you will not be able to see the login page. To see the status of the configuration, follow the instructions to connect the Data Transforms Instance as described in Connect to Oracle Data Transforms Instance. Check the

`transformsStartup.log` file, fix the issues, and repeat these steps to provision the repository.

If you want to check the `transformsStartup.log` file without connecting to the instance, run the **displayStartupLog** command from the OCI Console using the compute instance run command feature. See Use the OCI Console to Run Commands in the Data Transforms Instance for detailed instructions.

4
Access

Get information about the many ways you can use to connect to the Data Transforms compute instance.

Topics

- Connect to Oracle Data Transforms Instance
  This section includes information about connecting to the Data Transforms compute instance by using a Secure Shell (SSH) connection.
- Access the Oracle Data Transforms Compute Instance through VNC
  To access the Oracle Data Transforms instance through VNC, you need to install a VNC viewer on your local computer. You also need to provide the VNC password that you had provided during the stack creation.
- Access Oracle Data Transforms
  This topic provides information on how you can access the Oracle Data Transforms from a web browser.

4.1 Connect to Oracle Data Transforms Instance

This section includes information about connecting to the Data Transforms compute instance by using a Secure Shell (SSH) connection.

> **Note:**
>
> This section is applicable only for Advanced administrative users who would typically connect to the Data Transforms instance to examine the logs for errors. If you are a general Data Transforms developer, you can skip this section.

Topics:

- Connect to Oracle Data Transforms Instance from Linux
- Connect to Oracle Data Transforms Instance from Windows

4.1.1 Connect to Oracle Data Transforms Instance from Linux

You can connect to an Oracle Data Transforms compute instance by using a Secure Shell (SSH) connection. Most Linux distributions include an SSH client by default.

To connect to your Oracle Data Transforms compute instance from Linux,

1. To enable SSH access to Oracle user, log in to your instance using SSH as opc user.

   ```
   ssh opc@<IP Address>
   ```

Execute the following commands to copy the authorized keys to Oracle user:

```
sudo cp /home/opc/.ssh/authorized_keys /home/oracle/.ssh
sudo chown oracle:oracle /home/oracle/.ssh/authorized_keys
```

2. Use the following command to set the file permissions so that only you can read the file:

```
$ chmod 400 <private_key>
```

where `<private_key>` is the full path and name of the file that contains the private key associated with the instance you want to access.

3. Use the following SSH command to access the instance.

```
$ ssh -i <private_key> <username>@<public-ip-address>
```

where

- `<private_key>` is the full path and name of the file that contains the private key associated with the instance you want to access.
- `<username>` is the default name for the instance. The default user name is `oracle`.
- `<public-ip-address>` is your instance IP address that you retrieved from the Console.

4.1.2 Connect to Oracle Data Transforms Instance from Windows

You can connect to an Oracle Data Transforms compute instance by using a Secure Shell (SSH) connection. For Windows, you can download a free SSH client called PuTTY from http://www.putty.org.

To connect to your Oracle Data Transforms compute instance from Windows,

1. Open putty.exe.
2. In the Category pane, select Window, and then select Translation.
3. In the Remote character set drop-down list, select UTF-8. The default locale setting on Linux-based instances is UTF-8, and this configures PuTTY to use the same locale.
4. In the Category pane, select Session and enter the following:
   a. Host Name (or IP address):`<username>@<public-ip-address>`, where `<username>` is the default name for the instance. For Oracle Linux and CentOS images, the default user name is oracle. For the Ubuntu image, the default name is ubuntu and `<public-ip-address>` is your instance public IP address that you retrieved from the console.
   b. Port: 22
   c. Connection type: SSH
5. In the Category pane, expand Connection, expand SSH, and then click Auth.
6. Click Browse, and then select your private key.
7. Click Open to start the session.

If this is your first time connecting to the compute instance, you might see a message that the server's host key is not cached in the registry. Click **Yes** to continue the connection.

4.2 Access the Oracle Data Transforms Compute Instance through VNC

To access the Oracle Data Transforms instance through VNC, you need to install a VNC viewer on your local computer. You also need to provide the VNC password that you had provided during the stack creation.

To access Oracle Data Transforms utilities and logs through VNC, do the following:

1. Log in to the provisioned Oracle Data Transforms instance on Oracle Cloud Marketplace using SSH as opc user:

   ```
   ssh -i <private_key> opc@<IP Address>
   ```

2. Execute the following firewall commands to open the VNC ports:

   ```
   sudo firewall-cmd --permanent --new-service=odissh
   sudo firewall-cmd --permanent --service=odissh --set-description="ODI
   SSHserver"
   sudo firewall-cmd --permanent --service=odissh --add-port=5901-5905/tcp
   sudo firewall-cmd --permanent --add-service=odissh
   sudo firewall-cmd -reload
   ```

3. To create a VNC session, log in to the instance as Oracle user:

   ```
   ssh -i <private_key> oracle@<IP Address>
   ```

4. Start the VNC server by typing `vncserver`.
5. Enter the VNC password that you had provided during the stack creation.
6. Install a VNC viewer on your local computer.
7. Use SSH to connect to the compute instance running Oracle Data Transforms, as described in Connect to Oracle Data Transforms Instance.
8. On your local computer, connect to your instance and create a ssh tunnel for port 5901 (for display number 1):

   ```
   $ ssh -L 5901:localhost:5901 -i id_rsa oracle@<IP Address>
   ```

9. On your local computer, for the VNC to work, add an Ingress rule as follows:

   ```
   No 0.0.0.0/15 TCP All 5901 TCP traffic for ports: 5901
   ```

10. On your local computer, start a VNC viewer and establish a VNC connection to `localhost:1`.
11. Enter the VNC password that you had provided during the stack creation.
12. For connecting multiple users, after the vncpasswd utility exits, start the VNC server by typing `vncserver`. This will start a VNC server with display number 1 for the oracle user, and the VNC server starts automatically if your instance is rebooted. For example `vncserver@:2` or `vncserver@:3`.

> **Note:**
>
> Do not reboot the instance by choosing **Force reboot the instance by immediately powering off**. The VNC session will not start automatically.

To start developing your data loads, data flows or workflows,

1. In a web browser that has network connectivity to your compute instance, enter the url - `http://<ip address>:9999/oracle-data-transforms`.
2. Sign in to Data Transforms using the user and password supplied during instance deployment. Default user is SUPERVISOR.
3. Click **Connections** from the left pane of the **Home** page.
4. Click **Test connection**, to check if the created Autonomous Database connection is working.

4.3 Access Oracle Data Transforms

This topic provides information on how you can access the Oracle Data Transforms from a web browser.

You can access Oracle Data Transforms interface on Oracle Cloud Marketplace by:

- Navigating to Autonomous Database Data Tools Page
  For more details, refer to Access Oracle Data Transforms From Database Actions page.
- Click the link in OCI Stack Details page available in the **Application Information** tab.
- In a web browser that has network access to the Oracle Data Transforms Instance, type in `<ip address>:9999/oracle-data-transforms/` in the toolbar and click **Enter**.

You are navigated to the login page.

In the **Sign Into Oracle Data Integrator** page, enter the following login credentials and click **Connect**:

- Username - SUPERVISOR
- Password - Use the password that you provided during the ODI stack provisioning.

The Oracle Data Transforms interface appears allowing you to perform the data transformations. The very first screen you will see after you log in is the Home tab, which provides easy wizards to complete the following two actions:

- Load Data – Allows you to load multiple tables from the source connection schema to the target connection schema. You can also do this by clicking the required project tile in the Projects tab, clicking **Data Loads**, and then clicking **Create Data Load**.
- Transform Data - Allows you to create data flows to map data and run transformations. You can also do this by selecting an existing project and on the Project Details page, clicking **Create Data Flow**.
- Export - Allows you to export Data Transforms artifacts such as projects, connections, data loads, data flows, workflows, and schedules from one environment and import them to another.
- Import - Allows you to import the exported Data Transforms artifacts.

5

Connections

Learn about the connection types that Oracle Data Transforms supports and how to work with connections. You can also get instructions that are specific to certain connections.

Topics

5.1 Work with Connections

Connections help you to connect Data Transforms to various technologies reachable from your OCI network.

This section describes the generic steps to create a connection. The displayed connection detail options may vary depending on the selected connection type.

Apart from the connection types listed in Supported Connection Types you can create custom connectors, which you can use to connect Data Transforms to any JDBC supported data sources. See Create Custom Connectors.

To create a new connection:

1. From the left pane of the Home page, click the **Connections** tab.
   Connections page appears.
2. Click **Create Connection**.
   The **Create Connection** page slides in.
3. Do one of the following:
   - In the **Select Type** field, enter the name or part of the name of the connection type.
   - Select the type of connection that you wish to create.
     - Databases - Allows you to configure any connection type for supported database types.
     - Applications - Allows you to configure any connection type for supported applications.
     - Services - Allows you to configure any connection type for supported services.
4. After selecting the required connection type, click **Next**.
   The **Connection Name** field is pre-populated with a default name. You can edit this value.
5. For **Connection Details**, provide the connection details for the selected type such as:
   - Connection -
     - JDBC URL - The URL to connect to the data server. For example:
       `jdbc:weblogic:sqlserver://hostname:port[;property=value[;...]]`

       > **Note:**
       >
       > If you are accessing Data Transforms from an Autonomous Database on a public endpoint, then the allowed port numbers are 2484, 1521-1525, 1708, 3306, 80.

       For connectors that use an autoREST driver that provides the model files along with the driver, specify the servername and other properties required to connect to that datasource. For example:

       `jdbc:weblogic:autorest://servername;[property=value[;...]]`
     - User - The user name, if required, for connecting to the server.
     - Password - The password for connecting to server.
   - Advanced Options

- Array Fetch Size - When reading large volumes of data from a data server, Oracle Data Transforms fetches successive batches of records. This value is the number of rows (records read) requested by Oracle Data Transforms on each communication with the data server.
- Batch Update Size - When writing large volumes of data into a data server, Oracle Data Transforms pushes successive batches of records. This value is the number of rows (records written) in a single Oracle Data Transforms INSERT command.

> **Note:**
>
> Set Batch Update Size to 1000 or less for loading tables with BLOB data type columns.

- Degree of Parallelism for Target - This value indicates the number of threads allowed for a loading task. The default value is 1. The maximum number of threads allowed is 99.

> **Note:**
>
> Connection details are specific and the above options vary based on the selected connection type. For the default connection that is created during provisioning, only the User and Password fields are editable. All the other fields are disabled.

6. After providing all the required connection details, click **Test Connection** to test the connection.
   If the test connection fails, do one of the following:
   - Check whether the Autonomous Database from where you are accessing Data Transforms is configured to use a private endpoint.
     See Enable Access to Private Data Sources from Autonomous Database for more information.

     See Enable Access to Private Data Sources from Autonomous Database for more information.
   - If a connection between Autonomous Databases on a Private Endpoint fails with a "Mismatch with server cert DN" error, check whether you have specified the `ssl_server_dn_match=yes` property in the JDBC URL. See Troubleshoot Mismatch with Server Cert DN Error for more information.
7. Click **Create**.
   The new connection is created.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon (⋮) next to the selected connection to perform the following operations:

- Select **Edit** to edit the provided connection details.
- Select **Test Connection** to test the created connection.
- Click **Export** to export the connection. See Export Objects.
- Select **Delete Schema** to delete schemas.

- Select **Delete Connection** to delete the created connection.

You can also search for the required connection to know its details based on the following filters:

- **Name** of the connection.
- **Technology** associated with the created connection.

Topics

5.2 Supported Connection Types

This topic lists the connection types that are supported for connecting to Data Transforms.

Government cloud regions are required to support FIPS compliance standards for data protection. Data Transforms in Government realms is FIPS 140-2 Level 1 compliant. Make sure the connections sources are FIPS compliant to ensure secure communication between the servers.

> **Note:**
>
> APPLIES TO: Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.
>
> - For the connectors that require driver installation, you need to copy the jar files to the `/u01/oracle/transforms_home/userlibs` directory before you add the connection.
> - Apart from the connection types listed here you can create custom connectors, which you can use to connect Data Transforms to any JDBC supported data sources. See Create Custom Connectors.

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Aha!	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:servername=https://{yourdomain}.aha.io; AuthenticationMethod=OAuth2; clientid=**********;clientsecret=**********; authuri=https://{yourdomain}.aha.io/oauth/authorize; tokenuri=https://{yourdomain}.aha.io/oauth/token; redirectUri=http://localhost;`	
Ahrefs	Application	Yes	Yes	Yes	No	**URLParameter**: `jdbc:weblogic:autorest:ServerName=https://apiv2.ahrefs.com;AuthenticationMethod=URLParameter;SecurityToken=xxxxxxxx;AuthParam=token;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Amazon Aurora	Database	Yes	Yes	Yes	Yes	• **userIDPassword**: `jdbc:weblogic:postgresql://servername:port;AuthenticationMethod=userIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:postgresql://servername:port;AuthenticationMethod=kerberos;ServicePrincipalName=servicePrincipalName;[property=value[;...]]; User: [Ignored] Password: [Ignored]` • **EntraIDPassword**: `jdbc:weblogic:postgresql://servername:port;AuthenticationMethod=EntraIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **EntraIDServicePrincipal**: `jdbc:weblogic:postgresql://servername:port;AuthenticationMethod=EntraIDServicePrincipal; ServicePrincipal=your_service_principal ;Secret=your_client_secret; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Amazon EMR Hive	Database	Yes	Yes	Yes	No	• **delegationToken**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=delegationToken;DelegationToken=delegationToken;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **userIdPassword**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=userIdPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **none**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=none;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Amazon Redshift	Database	Yes	Yes	Yes	Yes	`jdbc:weblogic:redshift://server_name:port_number;DatabaseName=database_name;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Apache Hive	Database	Yes	Yes	Yes	No	• **delegationToken**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=delegationToken;DelegationToken=delegationToken; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **userIdPassword**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=userIdPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **none**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=none; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Apache Iceberg	Database	Yes	Yes	Yes	Yes	Authentication modes supported are None, Simple, and OAuth2.0. `http://<host>:<port>/iceberg`	For information about creating a connection using Apache Iceberg, see Create an Apache Iceberg Connection.

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Apache Impala	Database	Yes	Yes	Yes	No	• **userIdPassword**: `jdbc:weblogic:impala://servername:port;DatabaseName=databaseName;AuthenticationMethod=userIdPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:impala://servername:port;DatabaseName=databaseName;AuthenticationMethod=kerberos;ServicePrincipleName=serviceprincipalname;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **none**: `jdbc:weblogic:impala://servername:port;DatabaseName=databaseName;AuthenticationMethod=none;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Apache Spark SQL	Database	Yes	Yes	Yes	No	• **kerberos**: `jdbc:weblogic:sparksql://servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal; [property=value[;...]]; User: [Ignored] Password: [Ignored]` • **userIdPassword**: `jdbc:weblogic:sparksql://servername:port;DatabaseName=database;AuthenticationMethod=userIdPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]`	
AWS S3	Database	Yes	Yes	Yes	No	**AWS**: `jdbc:weblogic:autorest:AuthenticationMethod=AWS;servicename=s3;accesskey=*******;secretkey=*******;`	
Azure Billing	Application	Yes	Yes	Yes	No	`jdbc:weblogic:autorest://servername; [property=value[;...]];`	
Azure Compute	Database	Yes	Yes	Yes	No	`jdbc:weblogic:autorest://servername; [property=value[;...]]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Azure Data Lake Storage	Database	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest: ServerName=https://{accountName}.dfs.core.windows.net;AuthenticationMethod=OAuth2;ClientID=xxxxxxxx;ClientSecret=xxxxxxxx; RefreshToken=xxxxxxxx;Scope=https://storage.azure.com/user_impersonation offline_access;TokenURI=https://login.microsoftonline.com/{tenantID}/oauth2/v2.0/token;AuthorizationURI=https://login.microsoftonline.com/{tenantID}/oauth2/v2.0/authorize;RedirectURI: http://localhost;`	
Azure Reserved VM Instances	Database	Yes	Yes	Yes	No	`jdbc:weblogic:autorest://servername; [property=value[;...]];`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Azure Resource Health	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:ServerName=https://management.azure.com;AuthenticationMethod=OAuth2;TenantId=xxxxxxxx;ClientID=xxxxxxxx;ClientSecret=xxxxxxxx;AccessToken=xxxxxxxx;RefreshToken=xxxxxxxx;Scope=https://management.azure.com/.default offline_access;TokenURI=https://login.microsoftonline.com/{tenantId}/oauth2/token;RedirectURI=http://localhost/;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Azure SQL Database	Database	Yes	Yes	Yes	Yes	• **activeDirectoryPassword**: `jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=activeDirectoryPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **ActiveDirectoryServicePrincipal**: `jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ActiveDirectoryServicePrincipal;ServicePrincipal=clientID;Secret=clientSecret;[property=value[;...]]; User: [Ignored] Password: [Ignored]` • **auto**: `jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=auto;[property=value[;...]]; User: [Optional] Password: [Optional]` • **kerberos**: `jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipalName=service_principal_name;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
						[property=value[;...]]; User: [Ignored] Password: [Ignored] • **ntlm**: jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ntlm; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **ntlmjava**: jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ntlmjava; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **ntlm2java**: jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ntlm2java; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **userIdPassword**: jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=userIdPassword;	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
						[property=value[;...]]; User: [Mandatory] Password: [Mandatory]	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Azure Synapse Analytics	Database	Yes	Yes	Yes	Yes	• **activeDirectoryPassword**: `jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=activeDirectoryPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **ActiveDirectoryServicePrincipal**: `jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ActiveDirectoryServicePrincipal;ServicePrincipal=clientID;Secret=clientSecret;[property=value[;...]]; User: [Ignored] Password: [Ignored]` • **auto**: `jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=auto;[property=value[;...]]; User: [Optional] Password: [Optional]` • **kerberos**: `jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipalName=service_principal_name;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
						[property=value[;...]]; User: [Ignored] Password: [Ignored] • **ntlm**: jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ntlm; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **ntlmjava**: jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ntlmjava; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **ntlm2java**: jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ntlm2java; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **userIdPassword**: jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=userIdPassword;	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
						[property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
BigCommerce	Application	Yes	Yes	Yes	No	**HTTPHeader**: jdbc:weblogic:autorest:servername=https://api.bigcommerce.com/stores/{store_hash};AuthenticationMethod=HTTPHeader;securitytoken=**********;authheader=X-Auth-Token;	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Cassandra	Database	Yes	Yes	Yes	Yes	• **userIdPassword**: `jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=userIdPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal;[property=value[;...]]; User: [Ignored] Password: [Ignored]` • **none**: `jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=none;[property=value[;...]]; User: [Ignored] Password: [Ignored]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Cloudera CDH Hive	Database	Yes	Yes	Yes	No	• **delegationToken**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=delegationToken;DelegationToken=delegationToken;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **userIdPassword**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=userIdPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **none**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=none;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Confluence Cloud	Database	Yes	Yes	Yes	No	**Basic**: `jdbc:weblogic:autorest:servername=https://<your-domain.atlassian.net>;authenticationmethod=Basic;`	
Data Transforms	Service	Yes	Yes	Yes	No	-	For instructions on connecting to an existing Data Transforms instance, see Create a Data Transforms Connection for Remote Data Load.

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
DataStax	Application	Yes	Yes	Yes	Yes	• **userIdPassword**: `jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=userIdPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal;[property=value[;...]]; User: [Ignored] Password: [Ignored]` • **none**: `jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=none;[property=value[;...]]; User: [Ignored] Password: [Ignored]`	`jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;[property=value[;...]]` where `KeyspaceName` specifies the default name of the Cassandra keyspace to which the driver connects.

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Base	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Delta Share	Application	Yes	Yes	Yes	No	-	For instructions on creating a connection using Delta Share, see Create a Delta Share Connection

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
DocuSign	Database	Yes	Yes	Yes	No	• **userIdPassword**`jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=userIdPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal;[property=value[;...]]; User: [Ignored] Password: [Ignored]` • **none**: `jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=none;[property=value[;...]]; User: [Ignored] Password: [Ignored]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
eBay	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:ServerName=https://api.eBay.com;AuthenticationMethod=OAuth2;ClientID=NameHere-APIUserA-BCD-1234e5f6g-h7i8j890;ClientSecret=xxxxxxxx;AccessToken=xxxxxxxx;RefreshToken=xxxxxxxx;Scope=https://api.eBay.com/oauth/api_scopeTokenURI=https://api.eBay.com/identity/v1/oauth2/tokenRedirectURI=http://localhost/;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
EnterpriseDB	Database	Yes	Yes	Yes	Yes	• **userIDPassword**: `jdbc:weblogic:postgresql://servername:port;AuthenticationMethod=userIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:postgresql://servername:port;AuthenticationMethod=kerberos;ServicePrincipalName=servicePrincipalName;[property=value[;...]]; User: [Ignored] Password: [Ignored]` • **EntraIDPassword**: `jdbc:weblogic:postgresql://servername:port;AuthenticationMethod=EntraIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **EntraIDServicePrincipal**: `jdbc:weblogic:postgresql://servername:port;AuthenticationMethod=EntraIDServicePrincipal; ServicePrincipal=your_service_principal ;Secret=your_client_secret; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
FinancialForce	Application	Yes	Yes	Yes	Yes	• **userIDPassword**: `jdbc:weblogic:sforce://servername;AuthenticationMethod=userIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **oauth2.0**: `jdbc:weblogic:sforce://servername;AccessToken=your_access_token;AuthenticationMethod=oauth2.0; [property=value[;...]]; User: [Ignored] Password: [Ignored]`	
FourSquare	Application	Yes	Yes	Yes	No	• **URLParameter**: `jdbc:weblogic:autorest:servername=https://api.foursquare.com;AuthenticationMethod=URLParameter;securitytoken=**********; authparam=oauth_token;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Base	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Generic Rest	Application	Yes	Yes	Yes	No	-	For information about connecting to any REST service endpoint to create a connection, see Create a REST Server Connection.

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Generic Rest Config	Application	Yes	No	No	No	**OAuth2**: `jdbc:weblogic:autorest:ServerName=https://{accountName}.dfs.core.windows.net;AuthenticationMethod=OAuth2;ClientID=xxxxxxxx;ClientSecret=xxxxxxxx;RefreshToken=xxxxxxxx;Scope=https://storage.azure.com/user_impersonation offline_access;TokenURI=https://login.microsoftonline.com/{tenantID}/oauth2/v2.0/token;AuthorizationURI=https://login.microsoftonline.com/{tenantID}/oauth2/v2.0/authorize;RedirectURI: http://localhost;`	For information about connecting to any REST service endpoint to create a connection, see Create a REST Server Connection.
GitHub	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:clientId=123a4b567c8901234567d8901e234fg5;clientSecret=**********;authUri=https://github.com/login/oauth/authorize;AuthenticationMethod=OAuth2;tokenUri=https://github.com/login/oauth/access_token;redirectUri=http://localhost;authenticationMethod=OAuth2;refreshtoken=**********;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Google Ads	Application	Yes	No	No	Depends on the driver	`jdbc:weblogic:autorest://servername;[property=value[;...]];`	Requires driver installation

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Google AdSense	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:servername=https://adsense.googleapis.com/v2;clientId=123456789012-abc123def456ghi789jkl012mno345pq.apps.googleusercontent.com;clientSecret=**********;authUri=https://accounts.google.com/o/oauth2/auth;tokenUri=https://oauth2.googleapis.com/token;redirectUri=http://localhost;AuthenticationMethod=OAuth2;scope=https://www.googleapis.com/auth/adsense;refreshtoken=**********;`	
Google Analytics	Application	Yes	Yes	Yes	No	• **OAuth2**: `jdbc:weblogic:googleanalytics4:AddTables='{myTableDefinitionString}';ClientID=clientID;ClientSecret=client_secret;AuthenticationMethod=OAuth2;RefreshToken=refresh_token;Scope=scope; [property=value[;...]]; User: [Ignored] Password: [Ignored]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Google BigQuery	Database	Yes	Yes	Yes	No	• **oauth2**: `jdbc:weblogic:googlebigquery:Project=project;Dataset=dataset;AccessToken=accesstoken;RefreshToken=refreshtoken;ClientID=clientid;ClientSecret=clientsecret;AuthenticationMethod=oauth2; [property=value[;...]]; User: [Ignored] Password: [Ignored]` • **serviceaccount**: `jdbc:weblogic:googlebigquery:Project=project;Dataset=dataset;ServiceAccountEmail=serviceAccountEmail;ServiceAccountPrivateKey=serviceAccountPrivateKey.json; AuthenticationMethod=serviceaccount; [property=value[;...]]; User: [Ignored] Password: [Ignored]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Google Calendar	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:servername=https://www.googleapis.com/calendar/v3; clientId=123456789012-abc123def456ghi789jkl012mno345pq.apps.googleusercontent.com; clientSecret=**********; authUri=https://accounts.google.com/o/oauth2/auth; tokenUri=https://accounts.google.com/o/oauth2/token; redirectUri=http://localhost; AuthenticationMethod=OAuth2; scope=https://www.googleapis.com/auth/calendar https://www.googleapis.com/auth/calendar.readonly https://www.googleapis.com/auth/calendar.eventshttps://www.googleapis.com/auth/calendar.events.readonly https://www.googleapis.com/auth/calendar.settings.readonly https://www.googleapis.com/auth/calendar.addons.execute; authuri=https://accounts.google.com/o/oauth2/auth;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
						tokenuri=https://accounts.google.com/o/oauth2/token; refreshtoken=**********;	
Google Campaign Manager	Application	Yes	Yes	Yes	No	jdbc:weblogic:autorest://servername; [property=value[;...]];	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Google Contacts	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:servername=people.googleapis.com/v1; clientId=123456789012-abc123def456ghi789jkl012mno345pq.apps.googleusercontent.com; clientSecret=**********; authUri=https://accounts.google.com/o/oauth2/auth; tokenUri=https://accounts.google.com/o/oauth2/token; redirectUri=http://localhost; AuthenticationMethod=OAuth2; scope=https://www.googleapis.com/auth/contacts.other.readonly https://www.googleapis.com/auth/contactshttps://www.googleapis.com/auth/contacts.readonly https://www.googleapis.com/auth/directory.readonlyhttps://www.googleapis.com/auth/profile.agerange.read https://www.googleapis.com/auth/profile.emails.readhttps://www.googleapis.com/auth/profile.language.read https://`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
						www.googleapis.com/auth/ user.addresses.readhttps:// www.googleapis.com/auth/user.birthday.read https://www.googleapis.com/auth/ user.emails.readhttps://www.googleapis.com/ auth/user.gender.read https:// www.googleapis.com/auth/ user.organization.readhttps:// www.googleapis.com/auth/ user.phonenumbers.read https:// www.googleapis.com/auth/ userinfo.emailhttps://www.googleapis.com/ auth/userinfo.profile; refreshtoken: **********;	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Google Drive	Database	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:servername=https://www.googleapis.com/drive/v3;clientId=123456789012-abc123def456ghi789jkl.apps.googleusercontent.com;clientSecret=**********;authUri=https://accounts.google.com/o/oauth2/auth;tokenUri=https://accounts.google.com/o/oauth2/token;redirectUri=http://localhost;AuthenticationMethod=OAuth2;scope=https://www.googleapis.com/auth/drive;refreshtoken=**********;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Google Search Ads 360	Application	Yes	Yes	Yes	No	**OAuth2**: jdbc:weblogic:autorest:servername=https://www.googleapis.com/doubleclicksearch/; clientId: 372185870949-v1f1h94mvnad4hkajr4tonlia51vtj2n.apps.googleusercontent.com; clientSecret: **********; authUri: https://accounts.google.com/o/oauth2/auth; tokenUri: https://oauth2.googleapis.com/token; redirectUri: http://localhost;AuthenticationMethod:OAuth2; scope: https://www.googleapis.com/auth/doubleclicksearch; refreshtoken: **********;	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Greenplum	Database	Yes	Yes	Yes	No	• **userIdPassword**: `jdbc:weblogic:greenplum://servername:port;KeyspaceName=keyspace;AuthenticationMethod=userIdPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:greenplum://servername:port;DatabaseName=databaseName;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal;[property=value[;...]]; User: [Ignored] Password: [Ignored]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Hortonworks Hive	Database	Yes	Yes	Yes	No	• **delegationToken**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=delegationToken;DelegationToken=delegationToken;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **userIdPassword**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=userIdPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **none**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=none;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
HubSpot	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:hubspot:AuthenticationMethod=OAuth2;ClientID=<CLIENT_ID>;ClientSecret=<CLIENT_SECRET>;AccessToken=<ACCESS_TOKEN>;`	
Hypersonic SQL	Database	Yes	Yes	Yes	Yes	`jdbc:hsqldb:.;ifexists=true[;<property>=<value>...]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
IBM BigInsights	Database	Yes	Yes	Yes	No	• **delegationToken**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=delegationToken;DelegationToken=delegationToken; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **userIdPassword**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=userIdPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **none**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=none; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
IBM DB2 Hosted	Database	Yes	Yes	Yes	Yes	• **cleartext**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=cleartext;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **client**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=client;[property=value[;...]]; User: [Ignored] Password: [Ignored]` • **encryptedPassword**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=encryptedPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **encryptedPasswordAES**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=encryptedPasswordAES;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **encryptedUIDPassword**: `jdbc:weblogic:db2://`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
						servername:port;DatabaseName=database;AuthenticationMethod=encryptedUIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **encryptedUIDPasswordAES**: jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=encryptedUIDPasswordAES;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **kerberos**: jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal; [property=value[;...]]; User: [Ignored] Password: [Ignored] • **pluginSecurity**: jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=pluginSecurity;PluginName=pluginName; [property=value[;...]]; User: [Ignored] Password: [Ignored]	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
IBM DB2 UDB	Database	Yes	Yes	Yes	Yes	• **cleartext**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=cleartext;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **client**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=client;[property=value[;...]]; User: [Ignored] Password: [Ignored]` • **encryptedPassword**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=encryptedPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **encryptedPasswordAES**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=encryptedPasswordAES;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **encryptedUIDPassword**: `jdbc:weblogic:db2://`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
						servername:port;DatabaseName=database;AuthenticationMethod=encryptedUIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **encryptedUIDPasswordAES**: jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=encryptedUIDPasswordAES;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **kerberos**: jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal; [property=value[;...]]; User: [Ignored] Password: [Ignored] • **pluginSecurity**: jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=pluginSecurity;PluginName=pluginName; [property=value[;...]]; User: [Ignored] Password: [Ignored]	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
IBM DB2 Warehouse	Database	Yes	Yes	Yes	Yes	• **cleartext**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=cleartext;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **client**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=client;[property=value[;...]]; User: [Ignored] Password: [Ignored]` • **encryptedPassword**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=encryptedPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **encryptedPasswordAES**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=encryptedPasswordAES;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **encryptedUIDPassword**: `jdbc:weblogic:db2://`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
						servername:port;DatabaseName=database;AuthenticationMethod=encryptedUIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **encryptedUIDPasswordAES**: jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=encryptedUIDPasswordAES;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **kerberos**: jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal; [property=value[;...]]; User: [Ignored] Password: [Ignored] • **pluginSecurity**: jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=pluginSecurity;PluginName=pluginName; [property=value[;...]]; User: [Ignored] Password: [Ignored]	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
IBM DB2/400	Database	Yes	Yes	Yes	Yes	• **cleartext**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=cleartext;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **client**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=client;[property=value[;...]]; User: [Ignored] Password: [Ignored]` • **encryptedPassword**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=encryptedPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **encryptedPasswordAES**: `jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=encryptedPasswordAES;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **encryptedUIDPassword**: `jdbc:weblogic:db2://`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
						servername:port;DatabaseName=database;AuthenticationMethod=encryptedUIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **encryptedUIDPasswordAES**: jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=encryptedUIDPasswordAES;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **kerberos**: jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal; [property=value[;...]]; User: [Ignored] Password: [Ignored] • **pluginSecurity**: jdbc:weblogic:db2://servername:port;DatabaseName=database;AuthenticationMethod=pluginSecurity;PluginName=pluginName; [property=value[;...]]; User: [Ignored] Password: [Ignored]	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Informix	Database	Yes	Yes	Yes	No	• jdbc:weblogic:informix://servername:port;InformixServer=informixserver;Database=databasename;[property=value[;...]];User: [Mandatory] Password: [Mandatory] • jdbc:weblogic:informix://servername:port;InformixServer=informixserver;DatabaseName=databasename;AlternateServers=(alternateserver);ConnectionRetryCount=connectionretrycount;ConnectionRetryDelay=connectionretrydelay;[property=value[;...]];User: [Mandatory] Password: [Mandatory]	
Jira	Application	Yes	Yes	Yes	No	**Basic**: `jdbc:weblogic:autorest:servername=https://your-domain.atlassian.net;AuthenticationMethod=Basic;`	
Klaviyo	Application	Yes	Yes	Yes	No	**URLParameter**: `jdbc:weblogic:autorest:servername=https://a.klaviyo.com/api;AuthenticationMethod=URLParameter;securitytoken=*****;authparam=api_key;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Magento	Application	Yes	No	No	Depends on the driver	`jdbc:weblogic:autorest://servername;[property=value[;...]];`	Requires driver installation
Mailchimp	Application	Yes	Yes	Yes	No	**BearerToken**: `jdbc:weblogic:autorest:servername=us20.api.mailchimp.com;AuthenticationMethod=BearerToken;securitytoken=*******;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
MapR Hive	Database	Yes	Yes	Yes	No	• **delegationToken**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=delegationToken;DelegationToken=delegationToken;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **userIdPassword**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=userIdPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **none**: `jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=none;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Marketo	Application	Yes	Yes	Yes	No	• **OAuth2**: `jdbc:weblogic:autorest:servername=https://{api_id}.mktorest.com;AuthenticationMethod=OAuth2;clientid=a1234bc5-67d8-9e01-f23g-4h567ijk89l0;clientsecret=**********;tokenuri=https://{api_id}.mktorest.com/identity/oauth/token;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Microsoft Dynamics 365	Application	Yes	Yes	Yes	Yes	• **OAuth2**: `jdbc:weblogic:dynamics365:ServiceURL=serviceurl; AuthenticationMethod=OAuth2;ClientID=client_id; ClientSecret=client_secret;TokenURI=token_uri;RefreshToken=refresh_token; [property=value[;...]]; User: [Ignored] Password: [Ignored]` • **NTLM**: `jdbc:weblogic:dynamics365:ServiceURL=serviceurl; AuthenticationMethod=NTLM;NTLMDomain=nTLMDomain;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Microsoft Share Point	Application	Yes	Yes	Yes	Yes	**Basic**: `jdbc:weblogic:sharepoint:ServiceURL=service url; AuthenticationMethod=Basic;ClientID=client_id; ClientSecret=client_secret;TokenURI=token_uri;RefreshToken=refresh_token; [property=value[;...]]; User: [Ignored] Password: [Ignored]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Microsoft SQL Server	Database	Yes	Yes	Yes	Yes	• **activeDirectoryPassword**: `jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=activeDirectoryPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **ActiveDirectoryServicePrincipal**: `jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ActiveDirectoryServicePrincipal;ServicePrincipal=clientID;Secret=clientSecret;[property=value[;...]];User: [Ignored] Password: [Ignored]` • **auto**: `jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=auto;[property=value[;...]]; User: [Optional] Password: [Optional]` • **kerberos**: `jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipalName=service_principal_name;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
						[property=value[;...]]; User: [Ignored] Password: [Ignored] • **ntlm**: jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ntlm;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **ntlmjava**: jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ntlmjava;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **ntlm2java**: jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ntlm2java;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **userIdPassword**: jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=userIdPassword;	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
						[property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
MongoDB	Database	Yes	Yes	Yes	Yes	• **None**: jdbc:weblogic:mongodb://host:port;AuthenticationDatabase=auth_db; AuthenticationMethod=None; [property=value[;...]]; User: [Ignored] Password: [Ignored] • **UserIDPassword**: jdbc:weblogic:mongodb://host:port;AuthenticationDatabase=auth_db; AuthenticationMethod=UserIDPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **Plain**: jdbc:weblogic:mongodb://host:port;AuthenticationDatabase=auth_db; AuthenticationMethod=Plain; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] • **Kerberos**: jdbc:weblogic:mongodb://host:port;AuthenticationDatabase=auth_db; AuthenticationMethod=Kerberos; [property=value[;...]]; User: [Ignored] Password: [Ignored]	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
MySQL	Database	Yes	Yes	Yes	Yes	`jdbc:mysql://<host>[:<port>]/[<database>]?relaxAutoCommit=true&useCursorFetch=true[&<property>=<value>...]`	Make sure that the system variable property `sql_require_primary_key` is set to OFF. Otherwise, an ADW to MySQL mapping could fail with a “Table does not exist” error.

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI GoldenGate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
MySQL Heatwave	Database	Yes	Yes	Yes	Yes	`jdbc:mysql://<host>[:<port>]/[<database>]?relaxAutoCommit=true&useCursorFetch=true[&<property>=<value>...]`	If MySQL Heatwave database is created with high availability, then write operation is not supported. Make sure that the system variable property `sql_require_primary_key` is set to OFF. Otherwise, an ADW to MySQL Heatwave mapping could fail with a “Table does

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
							not exist" error.
Netezza	Database	Yes	No	No	Depends on the driver	-	Oracle Data Transforms uses the Netezza JDBC to connect to a NCR Netezza database. This driver must be installed in your Data Transforms `userlibs` directory. You can download the Netezza JDBC driver from the IBM website.

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Oracle	Database	Yes	Yes	Yes	Yes	• `jdbc:oracle:thin:@<host>:<port>:<sid>` • `jdbc:oracle:thin:@<host>:<port/ServiceName>` • `jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=<host>)(PORT=<port>))(CONNECT_DATA=(SERVICE_NAME=<Service Name>)))`	For Data Integrator Web Edition, write operation is supported only on Oracle cloud database targets. For details refer to the Oracle terms of use before deploying the image from OCI marketplace.
Oracle Analytics Cloud	Application	Yes	Yes	Yes	No	-	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Oracle Business Intelligence Cloud (BICC) Connector	Application	Yes	Yes	Yes	No	-	For information about creating a connection using Oracle Business Intelligence Cloud (BICC) Connector, see Create an Oracle Business Intelligence Cloud Connector Connection.
Oracle EBS	Application	Yes	Yes	Yes	Yes	`jdbc:oracle:thin:@<host>:<port>:<sid>`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Base	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Oracle ERP Cloud	Application	Yes	Yes	No	No		For information about creating a connection using Oracle ERP Cloud, see Create an Oracle Enterprise Resource Planning Cloud Connection.
Oracle Financials Cloud	Application	Yes	Yes	Yes	No	-	For information about creating a connection using Oracle Financials Cloud, see Create an Oracle Financials Cloud Connection.

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data base	Supported in Data Transforms built into OCI GoldenGate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Oracle Fusion ERP	Application	Yes	Yes	Yes	No	-	
Oracle Fusion Sales	Application	Yes	Yes	Yes	No	-	
Oracle Fusion Service	Application	Yes	Yes	Yes	No	-	
Oracle GoldenGate – OCI	Service	Yes	Yes	Yes	Yes	-	
Oracle Marketing Cloud	Application	Yes	Yes	Yes	Yes	`jdbc:weblogic:eloqua:Company=<company_id>;[;property=value[;...]]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Oracle NetSuite	Application	Yes	Yes	Yes	No	`jdbc:ns://{Server Host}:{Server Port};ServerDataSource={Server Data Source};`	For information about creating a connection using Oracle Netsuite, see Create and Use an Oracle NetSuite Connection.
Oracle Object Storage	Database	Yes	Yes	Yes	Yes	-	For information about creating a connection using Oracle Object Storage, see Create an Oracle Object Storage Connection.
Oracle PeopleSoft	Application	Yes	Yes	Yes	No	`jdbc:oracle:thin:@<host>:<port>:<sid>`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Oracle Sales Cloud	Application	Yes	Yes	Yes	No	`jdbc:weblogic:oraclesalescloud://<base_url>;WSCompressData=none`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Oracle Service Cloud	Application	Yes	Yes	Yes	No	`jdbc:weblogic:oracleservicecloud:loginHost=host;[property=value[;...]]`	When using multiple JDBC connections for Oracle Service Cloud in Oracle Data Transforms, ensure that each connection is uniquely identified to avoid conflict. To do this, add the `DatabaseName` property to the JDBC URL. For example, `jdbc:weblogic:oracleservicecloud:loginHost=h`

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
							ost;DatabaseName=<*UniqueValue*>; where <*UniqueValue*> is a unique value for each environment. For example, DevConnection or ProdConnection. This allows separate configuration files for each environment.
Oracle SIEBEL	Application	Yes	Yes	Yes	No	jdbc:oracle:thin:@<host>:<port>:<sid>	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
PayPal	Application	Yes	Yes	Yes	No	**Basic**: `jdbc:weblogic:autorest:ServerName=https://api-m.sandbox.paypal.com;AuthenticationMethod=Basic;`	
Pivotal HD	Database	Yes	Yes	Yes	No	-	
Pivotal HDB	Database	Yes	Yes	Yes	No	• **userIdPassword**: `jdbc:weblogic:greenplum://servername:port;KeyspaceName=keyspace;AuthenticationMethod=userIdPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:greenplum://servername:port;DatabaseName=databaseName;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal;[property=value[;...]];User: [Ignored] Password: [Ignored]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
PostgreSQL	Database	Yes	Yes	Yes	Yes	• **userIDPassword**: `jdbc:weblogic:postgresql://servername:port;AuthenticationMethod=userIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **kerberos**: `jdbc:weblogic:postgresql://servername:port;AuthenticationMethod=kerberos;ServicePrincipalName=servicePrincipalName; [property=value[;...]]; User: [Ignored] Password: [Ignored]` • **EntraIDPassword**: `jdbc:weblogic:postgresql://servername:port;AuthenticationMethod=EntraIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **EntraIDServicePrincipal**: `jdbc:weblogic:postgresql://servername:port;AuthenticationMethod=EntraIDServicePrincipal;ServicePrincipal=servicePrincipal;Secret=secret; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Qmetry	Application	Yes	Yes	Yes	No	**URLParameter**: `jdbc:weblogic:autorest:servername=https://qtmcloud.qmetry.com;AuthenticationMethod=URLParameter;securitytoken=*****;authparam=apiKey;`	
Quick Books Online	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:servername=https://sandbox-quickbooks.api.intuit.com/v3/company/+++++; clientId: AB8oHGUgVpXQnqKehrhBrnnIEQ7vNa7YI4UOr6LEX1Ja6dx0CL; clientSecret: **********; authUri: https://appcenter.intuit.com/connect/oauth2; tokenUri: https://oauth.platform.intuit.com/oauth2/v1/tokens/bearer; redirectUri: http://localhost; AuthenticationMethod: OAuth2; refreshtoken: **********;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Quick Books Payments	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:servername=https://sandbox.api.intuit.com; clientId=ABCd1efGH2Ijkl3mN4OPQrsTu5vWxyZa6bCdEFgHijk7LMnOpQ;clientSecret=**********; authUri=https://appcenter.intuit.com/connect/oauth2; tokenUri=https://oauth.platform.intuit.com/oauth2/v1/tokens/bearer;redirectUri=http://localhost; AuthenticationMethod=OAuth2;scope=com.intuit.quickbooks.payment;refreshtoken=**********;`	
Quora Ads	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:ServerName=api.quora.com/ads;AuthenticationMethod=OAuth2;ClientID=xxxxxxx;ClientSecret=xxxxxxxx;AccessToken=xxxxxxx;Scope=ads_read;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Sage	Application	Yes	Yes	Yes	No	**OAuth2**: jdbc:weblogic:autorest:servername=api.accounting.sage.com;clientId:**********;clientSecret:**********;authUri:https://www.sageone.com/oauth2/auth/central?filter=apiv3.1;tokenUri:https://oauth.accounting.sage.com/token,redirectUri: http://localhost;AuthenticationMethod: OAuth2; scope: full_access; refreshtoken: **********;	
Salesforce Chatter	Application	Yes	Yes	Yes	No	**OAuth2**: jdbc:weblogic:autorest:servername=progresscom4-dev-ed.my.salesforce.com/services/data/v52.0; AuthenticationMethod=OAuth2; clientid=3MVG9SOw8KERNN08IkuAFNRo0MZbipoL4zeEhQzdT4IrNnqW7_UBvm1K7vZ4Y_OzbNxFn0.nJSfVBdqQuh4IL; clientsecret=**********; refreshtoken=**********; scope=full refresh_token; authuri=https://login.salesforce.com/services/oauth2/authorize; tokenuri=https://login.salesforce.com/services/oauth2/token;	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Salesforce.com	Application	Yes	Yes	Yes	Yes	• **userIDPassword**: `jdbc:weblogic:sforce://servername;AuthenticationMethod=userIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **OAuth2**: `jdbc:weblogic:sforce://servername;AccessToken=your_access_token;AuthenticationMethod=userIDPassword;[property=value[;...]]; User: [Ignored] Password: [Ignored]`	
SAP BW/4HANA	Database	Yes	Yes	Yes	No	• **Basic**: `jdbc:weblogic:s4hana:ServerName=servername;AuthenticationMethod=Basic;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **HTTPHeader**: `jdbc:weblogic:s4hana:ServerName=servername;AccessToken=your_access_token;AuthenticationMethod=HTTPHeader;AuthHeader=api_key;SecurityToken=security_token;[property=value[;...]]; User: [Ignored] Password: [Ignored]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
SAP HANA	Application	Yes	Yes	Yes	No	• **Basic**: `jdbc:weblogic:s4hana:ServerName=servername;AuthenticationMethod=Basic; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **HTTPHeader**: `jdbc:weblogic:s4hana:ServerName=servername;AccessToken=your_access_token;AuthenticationMethod=HTTPHeader;AuthHeader=api_key;SecurityToken=security_token; [property=value[;...]]; User: [Ignored] Password: [Ignored]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
SAP NetWeaver	Database	Yes	Yes	Yes	No	• **Basic**: `jdbc:weblogic:s4hana:ServerName=servername;AuthenticationMethod=Basic; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **HTTPHeader**: `jdbc:weblogic:s4hana:ServerName=servername;AccessToken=your_access_token;AuthenticationMethod=HTTPHeader;AuthHeader=api_key;SecurityToken=security_token; [property=value[;...]]; User: [Ignored] Password: [Ignored]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
SAP S/4HANA Cloud	Application	Yes	Yes	Yes	No	• **Basic**: `jdbc:weblogic:s4hana:ServerName=servername;AuthenticationMethod=Basic; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **HTTPHeader**: `jdbc:weblogic:s4hana:ServerName=servername;AccessToken=your_access_token;AuthenticationMethod=HTTPHeader;AuthHeader=api_key;SecurityToken=security_token; [property=value[;...]]; User: [Ignored] Password: [Ignored]`	
Semrush	Application	Yes	Yes	Yes	No	`jdbc:weblogic:autorest://servername; [property=value[;...]];`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
ServiceNow	Service	Yes	Yes	Yes	No	**OAuth2**:jdbc:weblogic:autorest:servername=instance.service-now.com/api; clientId=123a4b567c8901234567d8901e234fg5;clientSecret=**********; authUri=https://instance.service-now.com/oauth_auth.do; tokenUri=https://instance.service-now.com/oauth_token.do;redirectUri=http://localhost; AuthenticationMethod=OAuth2;refreshtoken=**********;	
Shopify	Application	Yes	Yes	Yes	No	jdbc:weblogic:autorest://servername; [property=value[;...]];	
Snowflake	Database	Yes	Yes	Yes	Yes	**Basic**: jdbc:weblogic:snowflake:AccountName=account_name;DatabaseName=database_name;Schema=schema_name;Warehouse=warehouse_name; [property=value[;...]];	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Square	Application	Yes	Yes	Yes	No	**BearerToken**: `jdbc:weblogic:autorest:servername=sandbox.api.intuit.com;AuthenticationMethod:BearerToken;securitytoken=**********;`	
Stripe	Application	Yes	Yes	Yes	No	**BearerToken**: `jdbc:weblogic:autorest:servername=https://api.stripe.com;AuthenticationMethod=BearerToken;securitytoken=**********;`	
SAP Sybase ASE	Database	Yes	Yes	Yes	Yes	`jdbc:weblogic:sybase://hostname:port[;property=value[;...]]`	
SAP Sybase IQ	Database	Yes	Yes	Yes	Yes	`jdbc:weblogic:sybase://hostname:port[;property=value[;...]]`	
Sybase As Anywhere	Database	Yes	Yes	Yes	Yes	`jdbc:weblogic:sybase://hostname:port[;property=value[;...]]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
TeamCity	Application	Yes	Yes	Yes	No	**BearerToken**: `jdbc:weblogic:autorest:ServerName=https://{your_teamcity_servername}.com;AuthenticationMethod=BearerToken;SecurityToken=Ab12cdEfG3HIJkL.MNOpQRStuvWX4yZabcD56cD7eF8GHiJ90Klm.nOpqRst2u3VWxYZabC4DEfGhIjk;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Teradata	Database	Yes	No	No	Depends on the driver	`jdbc:weblogic:teradata://<host>:<port>/<database>[;property=value[;...]]`	Data Transforms uses the Teradata JDBC Driver to connect to a Teradata Database. To use Teradata as a data source the Teradata Gateway for JDBC must be running, and this driver must be installed in your Data Transforms `userlibs` directory. You can download the JDBC driver from the

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transformsbuilt into Autonomous Database	Supported in Data Transformsbuilt into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
							Teradata website.

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Teradata 17+	Database	Yes	No	No	Depends on the driver	-	Data Transforms uses the Teradata JDBC Driver to connect to a Teradata Database. To use Teradata as a data source the Teradata Gateway for JDBC must be running, and this driver must be installed in your Data Transforms `userlibs` directory. You can download the JDBC driver from the

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Warehouse	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
							Teradata website..
Tumblr	Application	Yes	Yes	Yes	No	`jdbc:weblogic:autorest://servername; [property=value[;...]]`	
Veeva CRM	Application	Yes	Yes	Yes	Yes	• **userIDPassword**: `jdbc:weblogic:sforce://servername;AuthenticationMethod=userIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]` • **oauth2.0**: `jdbc:weblogic:sforce://servername;AccessToken=your_access_token;AuthenticationMethod=oauth2.0; [property=value[;...]]; User: [Ignored] Password: [Ignored]`	
Volusion	Application	Yes	Yes	Yes	No	`jdbc:weblogic:autorest://servername; [property=value[;...]]`	
Wistia	Application	Yes	Yes	Yes	No	**Basic**: `jdbc:weblogic:autorest:servername=https://api.wistia.com;authenticationmethod=Basic;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
WooCommerce	Application	Yes	No	No	Depends on the driver	`jdbc:weblogic:autorest://servername;[property=value[;...]];`	Requires driver installation

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
WordPress	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:servername=public-api.wordpress.com/rest/v1.1;AuthenticationMethod=OAuth2;clientid=*****;clientsecret=*********;accesstoken=*********;scope=global;oauthcode=**********;authuri=https://public-api.wordpress.com/oauth2/authorize;tokenuri=https://public-api.wordpress.com/oauth2/token;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Data Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Workday	Application	Yes	No	No	Depends on the driver	`jdbc:weblogic:autorest://servername; [property=value[;...]];`	Requires driver installation
X	Application	Yes	Yes	Yes	No	-	
Xero	Application	Yes	Yes	Yes	No	`jdbc:weblogic:autorest://servername; [property=value[;...]]`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Yelp	Application	Yes	Yes	Yes	No	**BearerToken**: `jdbc:weblogic:autorest:servername=https://api.yelp.com/v3;AuthenticationMethod=BearerToken;securitytoken=**********;`	
Zendesk	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:servername=https://example.zendesk.com;AuthenticationMethod=OAuth2;clientid=client_id_zendesk; clientsecret=********; scope=read write;authuri=/oauth/authorizations/new; tokenuri=/oauth/tokens;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Zoho CRM	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:AuthenticationMethod=OAuth2;clientid=a1234bc567d89e01f23g; clientsecret: **********;scope=ZohoCRM.settings.ALL,ZohoCRM.modules.ALL,ZohoCRM.notifications.ALL,ZohoCRM.org.READ,ZohoCRM.users.READ,ZohoCRM.settings.roles.READ,ZohoCRM.settings.profiles.READ,ZohoCRM.modules.Contacts.READ, ZohoSearch.securesearch.READ; authuri=https://accounts.zoho.com/oauth/v2/auth;tokenuri=https://accounts.zoho.com/oauth/v2/token;redirecturi=http://localhost; refreshToken=**********;`	

Name	Type	Supported in Data Integrator: Web Edition	Supported in Data Transforms built into Autonomous Database	Supported in Data Transforms built into OCI Golden Gate	Supports Write Operation	Authentication Mode: Connection URL Example	Notes
Zoom	Application	Yes	Yes	Yes	No	**OAuth2**: `jdbc:weblogic:autorest:servername=https://api.zoom.us/v2;clientId=0aBcDeF_GhI2j_KlMnOpQr;clientSecret=**********;authUri=https://zoom.us/oauth/authorize;tokenUri=https://zoom.us/oauth/token;redirectUri=http://localhost;AuthenticationMethod=OAuth2;accesstoken:**********;`	

5.3 Create Custom Connectors

APPLIES TO: Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

The Custom Connections page of the Administration tab of Oracle Data Transforms helps you to create custom connectors that point to any JDBC supported data sources.

The custom connectors will be listed in the Create Connection page where you can use them to connect data sources to Data Transforms. See Work with Connections for more information. To create a new connector:

1. In the left pane, click **Administration**.

A warning message appears.

2. Click **Continue**.
3. In the left pane, click **Custom Connections**.
   Custom Connections screen appears.
4. Click **Create Connection Type**.
   The Create Connection Type page appears.
5. From the **Category** drop-down select the type of connection that you wish to create whether database, application, or service.
6. Enter a name for the connection.
7. Enter the name of the JDBC Driver of the source connection. For example, `oracle.jdbc.OracleDriver`.

> **Note:**
>
> For connectors that require driver installation, you need to copy the jar files to the `/u01/oracle/transforms_home/userlibs` directory before you add the connection.

8. Click **OK**.

The newly created custom connection appears in the list and are available in the Create Connection page.

Click the **Actions** icon (⋮) next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Click **Export** to export the connection. See Export Objects.
- Select **Delete**, to delete the created connection.

> **Note:**
>
> You cannot delete custom connectors that have existing connections.

5.4 Create a Data Transforms Connection for Remote Data Load

You can connect to an existing Data Transforms instance and run a data load remotely.

To create this connection, you need to specify the URL of the Data Transforms instance along with the name of the ODI rest API from where you want to run the data load.

To define a Data Transforms connection:

1. From the left pane of the Home page, click the **Connections** tab.

   Connections page appears.
2. Click **Create Connection**.

   Create Connection page slides in.

3. For **Select Type**,
   - In the **Name** field, enter the name of the newly created connection
   - Select **Services** as the type of connection that you wish to create.
4. In the **Endpoint URL** textbox, enter the URL of the ODI rest API from where you want to run the data load. Enter the URL in the format `http://<host-ip-address>:<port>/odi-rest`.
5. In the **User** text box enter SUPERVISOR as the user name.
6. In the **Password** text box enter the ODI Supervisor password.
7. After providing all the required connection details, click **Test Connection** to test the established connection.
8. Click **Create**.

   The new connection is created.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon (⋮) next to the selected connection to perform the following operations:

- Select **Edit** to edit the provided connection details.
- Select **Test Connection** to test the created connection.
- Click **Export** to export the connection. See Export Objects.
- Select **Delete Schema** to delete schemas.
- Select **Delete Connection** to delete the created connection.

You can also search for the required connection to know its details based on the following filters:

- **Name** of the Connection.
- **Technology** associated with the created Connection.

5.5 Create an Apache Iceberg Connection

Apache Iceberg is an open standard table format that is optimized to manage large analytic datasets. Data Transforms supports the use of Apache Iceberg as a target to load data from any SQL-based data sources.

Data Transforms supports Oracle Object Storage (S3 compatibility) and AWS S3 storage services to store the parquet files for the Apache Iceberg tables.

The Data Transforms Apache Iceberg Connector requires that a REST Catalog already exists. This REST Catalog is setup based on Apache Gravitino (external link) with Iceberg Open API specification.

> **Note:**
>
> Data Transforms supports the use of Apache Gravitino version 0.7.0-incubating or lower to bring up the REST service.

This topic has the following sections:

Creating an Apache Iceberg Connection

You can configure an Apache Iceberg connection with the Iceberg REST Catalog by providing the REST URL and authentication details such as the username and password. You can also use the more secure OAuth 2.0 authentication to create the connection.

To create an Apache Iceberg connection:

1. From the left pane of the Home page, click the **Connections** tab.
   Connections page appears.
2. Click **Create Connection**.
   Create Connection page slides in.
3. Do one of the following:
   - In the **Select Type** field, enter the name or part of the name of the connection type.
   - Select the **Databases** tab.
4. Select **Apache Iceberg** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. In the **Catalog Name** textbox, enter a name.
8. In the **Rest URL** textbox, enter the URL of the REST server. Enter the value in the `<host>:<port>/<ServiceName>/iceberg` format.
9. From the **Authentication** drop-down section, do one of the following:
   - Select **None**.
   - Select **Simple** and enter the **Rest User** and **Rest Password**.
   - Select **OAuth** and enter the following details:
     - **Warehouse Location**: The location where you want to store the data. For example, `s3://my-bucket/my/table/location`
     - **Token URI**: The URL to obtain the OAuth Token in the format `http://<host>:<port>`
     - **Token Path**: The path to the OAuth token. For example, `/oauth2/token`.
     - **Client ID**: The OAuth Client ID.
     - **Client Secret**: The OAuth Client secret.
     - **Auth Scope**: The permissions granted to a client when accessing the Gravitino server. For example, a `test` Auth Scope value might indicate that the client is authorized to access resources related to the test" scope within Gravitino. [Optional]
     - **Grant Type**: The method that the authorization server should use to issue the access token. For example, `client_credentials` and `authorization_code`. [Optional]
10. Click **Test Connection**, to test the established connection.
11. After providing all the required connection details, click **Create**.

The Apache Iceberg connection is configured with REST Catalog, which stores the Iceberg data in Oracle Object Storage.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon (⋮) next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Select **Test Connection**, to test the created connection.
- Click **Export** to export the connection. See Export Objects.
- Select **Delete Schema**, to delete schemas.
- Select **Delete Connection**, to delete the created connection.

You can also search for the required Connection to know its details based on the following filters:

- **Name** of the Connection.
- **Technology** associated with the created Connection.

Creating and Running an Apache Iceberg Data Load

You can create a data load for any SQL-based source data source, such as Oracle, to load data into Apache Iceberg target tables. To use Apache Iceberg as a target data source, you need to provide the name of the connection and the namespace. A namespace in Apache Iceberg is similar to schema in relational databases.

After you create the data load, all the tables in the source schema are listed on the Data Load Detail page along with options to incrementally load, append, and merge the data for each of the selected source tables. When the data load run completes, you can read the data from the Iceberg tables. You can add the data load as a step in a workflow and create a schedule to run the workflows at a predefined time interval. See Create a New Workflow.

To create and run an Apache Iceberg Data Load:

1. Do one of the following:
   - On the Home page, click **Load Data**. The Create Data Load wizard appears.
     In the Create Data Load tab, enter a name if you want to replace the default value, add a description, and select a project from the drop-down.
   - On the Home page, click **Projects**, and then the required project tile. In the left pane, click **Data Loads**, and then click **Create Data Load**. The Create Data Load wizard appears.
2. Enter a name if you want to replace the default value and add a description.
3. For Load Processing do one of the following:
   - Select the **Internal** radio button and from the **Deployment Type** drop-down select **Data Transforms (Batch)**.
   - Select the **Delegate** radio button and from the **Deployment Type** drop-down select OCI GoldenGate. From the GoldenGate Deployment Connection select a connection.
4. Click **Next**.
5. In the Source Connection tab,
   a. From the **Connection Type** drop-down, select a SQL-based data source.

b. from the **Connection** drop-down, select the required connection from which you wish to add the data entities.

c. Click **Next**.

6. In the Target Connection tab,

   a. From the **Connection Type** drop-down, select **Apache Iceberg** as the connection type.

   b. From the **Connection** drop-down, select the connection you want to use to load the data into.

   c. Specify the **Namespace**. You can either select from existing namespaces or create a new namespace.

   d. Click **Save**.

   The Data Load Detail page appears listing all the source tables.

7. Select the required tables to load and the corresponding data load operation. The data load options you can use are Incremental Merge, Incremental Append, Append, and Do Not Load.

8. Click to save the changes. A green checkmark ( ) in the row indicates that the changes are saved.

9. Click to run the data load.
   A confirmation prompt appears when the data load starts successfully.

To check the status of the data load, see the Data Load Status panel on the right below the Target Schema details. For details about the panel, see Monitor Status of Data Loads, Data Flows, and Workflows. This panel shows links to the jobs that execute to run this data load. Click the link to monitor the progress on the Job Details page. For more information about jobs, see Create and Manage Jobs.

All the loaded tables along with their details are listed in the Data Entities page. To view the statistics of the data entities, click the **Actions** icon ( ) next to the data entity, click **Preview**, and then select the **Statistics** tab. See View Statistics of Data Entities for information.

5.6 Create a Delta Share Connection

Databricks Delta Share is an open protocol for secure data sharing. Oracle Data Transforms integrates with Delta Share to load data to Oracle Autonomous Database. You can use the Delta Share connection to load data from Databricks or Oracle Data Share.

To use Databricks as a source, you need to specify the URL of the Delta Sharing server along with the bearer token that lets you access the Delta Lake share server. To use Oracle Data Share as a source, you need to specify the URL for the token end point along with a client ID and the secret key.

This topic has the following sections:

- Creating the Delta Share Connection
- Creating and Running a Delta Share Data Load

Creating the Delta Share Connection

To define a Delta Share connection:

1. From the left pane of the Home page, click the **Connections** tab.
   Connections page appears.
2. Click **Create Connection**.
   Create Connection page slides in.
3. Do one of the following:
   - In the **Select Type** field, enter the name or part of the name of the connection type.
   - Select the **Databases** tab.
4. Select **Delta Share** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. In the **Share Endpoint URL** textbox, enter the URL of the Delta Sharing server. Enter the value in the `<host>:<port>/<shareEndpoint>/` format.
8. In the Connection section, do one of the following:
   - Select **Oracle Data Share** and provide the **Token Endpoint URL**, **Client ID**, and **Client Secret** for accessing the share.
     You can get this information from the Delta Share Profile JSON document that you will need to download from supplied to you by the Share Provider. (This is also where they get the Share Endpoint URL from)

     You can get this information from the Delta Share Profile JSON document that you can download from the activation link that is provided by the Data Share provider to access their share.
   - Select **Databricks** and in the **Bearer Token** text box enter the token for connecting to the Delta Sharing server.
9. If you need to use a proxy to access the Delta Share Server or Delta Share Storage configure the following settings:
   - In the **Proxy Host** textbox, enter the host name of the proxy server to be used for the connection.
   - In the **Proxy Port** textbox, enter the port number of the proxy server.
   - Select the following checkboxes depending on where the proxy is required:
     - **Use Proxy to access Delta Share Server**
     - **Use Proxy to access Delta Share Storage**
10. Click **Test Connection**, to test the established connection.
11. After providing all the required connection details, click **Create**.
    The new connection is created.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon (⋮) next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Select **Test Connection**, to test the created connection.
- Click **Export** to export the connection. See Export Objects.
- Select **Delete Schema**, to delete schemas.

- Select **Delete Connection**, to delete the created connection.

You can also search for the required Connection to know its details based on the following filters:

- **Name** of the Connection.
- **Technology** associated with the created Connection.

Creating and Running a Delta Share Data Load

To load data from Delta Share into Oracle Autonomous Database, the Oracle connection user must be an Admin user. Admin privileges are required so that the Oracle user can create and insert data into tables in another schema.

When you run the data load, Data Transforms loads the data onto a corresponding table in the target schema. The data load runs incrementally. The very first time you run a data load, Data Transforms copies all the data into new tables. For every subsequent data load run, it only uploads the changes. Any additions or deletions in the records will reflect in the target tables. Note that if there is any metadata change in the table, for example a column is added, Data Transforms creates a new table to load the data on to the target server. You could create a workflow, add the data load as a step, create a schedule to run the workflows at a predefined time interval. See Create a New Workflow.

To create and run a Delta Share data load:

1. Do one of the following:
   - On the Home page, click **Load Data**. The Create Data Load wizard appears.
     In the Create Data Load tab, enter a name if you want to replace the default value, add a description, and select a project from the drop-down.
   - On the Home page, click **Projects**, and then the required project tile. In the left pane, click **Data Loads**, and then click **Create Data Load**. The Create Data Load wizard appears.
     In the Create Data Load tab, enter a name if you want to replace the default value and add a description.
2. Click **Next**.
3. In the Source Connection tab,
   a. From the **Connection Type** drop-down, select **Delta Share**.
   b. from the **Connection** drop-down, select the required connection from which you wish to add the data entities.
   c. Select the share that you want to load tables from the **Share** drop-down. The drop-down lists all the shares for the selected connection.
   d. Click **Next**.
4. In the Target Connection tab,
   a. From the **Connection Type** drop-down, select **Oracle** as the connection type.

      > **Note:**
      >
      > This drop-down lists only JDBC type connections.

   b. From the **Connection** drop-down, select the required connection from to you wish to load the data entities.

c. Enter a unique name in the **Schema** textbox.

d. Click **Save**.

The Data Load Detail page appears listing all the tables in the selected share with their schema names.

> **Note:**
>
> For Delta Share data loads the Data Load Detail page only includes the option. You cannot apply different actions - incremental merge, incremental append, recreate, truncate, append - on the data entities before loading it to the target schema. This is to make sure that the data is consistent between the Delta Sharing server and the target schema.

5. Click to run the data load.
   A confirmation prompt appears when the data load starts successfully.

To check the status of the data load, see the Status panel on the right below the Target Schema details. For details about the Status panel, see Monitor Status of Data Loads, Data Flows, and Workflows. This panel shows links to the jobs that execute to run this data load. Click the link to monitor the progress on the Job Details page. For more information about jobs, see Create and Manage Jobs.

All the loaded data entities along with their details are listed in the Data Entities page. To view the statistics of the data entities, click the **Actions** icon ( ) next to the data entity, click **Preview**, and then select the **Statistics** tab. See View Statistics of Data Entities for information.

5.7 Create an Oracle Business Intelligence Cloud Connector Connection

Oracle Business Intelligence Cloud Connector (BICC) allows you to extract business data from a data source and load it into Autonomous Database.

To create an Oracle BICC connection you need to first configure external storage using the OCI Object Storage Connection tab in the BICC Console. You need to specify these connection details when you define the connection in Oracle Data Transforms.

You can use the BICC connection to choose the offerings whose data stores you want to extract. Data Transforms uses an Oracle Object Storage Data Server used by Oracle BICC to stage the extracted files, which you can then use as a source for mapping. Note that you cannot use an Oracle BICC connection as a target for mapping.

To define an Oracle BICC connection,

1. From the left pane of the Home page, click the **Connections** tab.

   Connections page appears.

2. Click **Create Connection**.

   Create Connection page slides in.

3. Do one of the following:

- In the **Select Type** field, enter the name or part of the name of the connection type.
- Select the **Applications** tab.

4. Select **Oracle BI Cloud Connector** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. Enter the URL in the **BI Cloud Connector Service URL** textbox.
8. In the **Connection** section, enter the following details:
   - In the **User** text box enter the user name configured in the Oracle BI Cloud Connector Console.
   - In the **Password** text box enter the password configured in the Oracle BI Cloud Connector Console.
9. In the **Storage** section, enter the following details:
   - In the **External Storage BICC Name** text box enter the name of the external storage as it appears in the Oracle BI Cloud Connector Console.
   - In the **External Storage Bucket** text box specify the bucket into which extracts are uploaded. Bucket names are obtained in the OCI Console.
   - In the **External Storage Name Space** text box specify the namespace. Namespace is obtained in the OCI Console.
   - In the **External Storage Region** text box enter the OCI Object Storage region.
   - In the **External Storage User** text box enter your Oracle Cloud Infrastructure username.
   - In the **External Storage Token** text box enter the auth token.
10. Click **Test Connection** to test the established connection.
11. Click **Create**.

    The new connection is created.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon (⋮) next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Select **Test Connection**, to test the created connection.
- Click **Export** to export the connection. See Export Objects.
- Select **Delete Schema**, to delete schemas.
- Select **Delete Connection**, to delete the created connection.

You can also search for the required Connection to know its details based on the following filters:

- **Name** of the Connection.
- **Technology** associated with the created Connection.

5.8 Create and use an Oracle Cloud Infrastructure Generative AI Connection

Oracle Cloud Infrastructure (OCI) Generative AI enables organizations to automate text summarization and dynamic content generation. Data Transforms integrates with OCI Generative AI to support the use of embedding vectors in a data flow.

Before you create an OCI Generative AI connection you create an Oracle Database 23ai connection. Data Transforms will use this connection to test the OCI Generative AI connection. To create the OCI Generative AI connection you need to specify details such as the OCI URL, User OCID, Tenancy OCID, Compartment OCID, Private Key, and Fingerprint information.

See Use Text Embedding Vector in a Data Flow for information on how you will use this connection to add vector embedding in a data flow.

To define Oracle Cloud Infrastructure Generative AI Connection:

1. From the left pane of the Home page, click the **Connections** tab.
   Connections page appears.
2. Click **Create Connection**.
   Create Connection page slides in.
3. For **Select Type**,
   - In the **Name** field, enter the name of the newly created connection.
   - Select **Services** as the type of connection.
4. Select **OCI Generative AI**, and **Next**.
5. For **Connection Details**, provide the following details:
   - **OCI URL** - The endpoint URL of the OCI Generative AI service.
   - **User OCID** - The user OCID from the Oracle Cloud Infrastructure Console.
   - **Tenancy OCID** - The tenancy OCID from the Oracle Cloud Infrastructure Console.
   - **Compartment OCID** - The compartment OCID from the Oracle Cloud Infrastructure Console.
   - **Private Key** - The private key in the PEM format. Specify the path to your downloaded private key file.
   - **Fingerprint** - The fingerprint of the key that was just added.
6. After proving all the required connection details, click **Create**.
   The new connection is created.
7. Click **Test Connection**, to test the established connection.
   A pop-up appears listing the Oracle Database 23ai connections that you have configured. Select the option you want to use to test this connection.

> **Note:**
>
> If test connection fails see Troubleshooting OCI Generative AI connection issues for instructions to fix the issue.

The newly created connection is displayed in the Connections page.

Troubleshooting OCI Generative AI connection issues

If **Test Connection** fails for an OCI Generative AI connection, do the following to troubleshoot the issue:

1. Make sure that you have entered all the connection information correctly. For example,
   - OCI URL sample: `https://inference.generativeai.us-chicago-1.oci.oraclecloud.com/20231130/actions/embedText`
   - Private Key sample: `MIIEvg.....beE/`
2. When you click **Test Connection**, you will be asked to choose an Oracle Connection. Make sure that it is an Oracle 23ai connection.
3. Log into that Oracle 23ai database as admin and run the following query to make sure the statuses are valid:

```
SELECT object_name, object_type, status
FROM dba_objects
WHERE object_name = 'DBMS_VECTOR_CHAIN' AND owner = 'CTXSYS';
```

4. Check whether the following plsql block works in your database. Data Transforms uses the plsql block to run the **Test Connection**.

```
exec dbms_vector_chain.drop_credential('OCI_CRED');
declare
  jo json_object_t;
begin
  -- create an OCI credential
  jo := json_object_t();
  jo.put('user_ocid', '<your user ocid>');
  jo.put('tenancy_ocid', '<your tenancy ocid');
  jo.put('compartment_ocid', '<your compartment ocid');
  jo.put('private_key', '<your private key');
  jo.put('fingerprint', '<your fingerprint>');
  dbms_output.put_line(jo.to_string);
  dbms_vector_chain.create_credential(
    credential_name => 'OCI_CRED',
    params => json(jo.to_string));
end;


select dbms_vector_chain.utl_to_embedding('hello', JSON('{"provider":
"ocigenai","credential_name" : "OCI_CRED", "url": "https://
inference.generativeai.us-chicago-1.oci.oraclecloud.com/20231130/actions/
embedText","model": "cohere.embed-english-light-v2.0"}')) from dual;
```

If you get an `HTTP request failed` error when running the plsql block query, try to grant network access to your user:

```
BEGIN
    DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
        host => '*',
        ace => xs$ace_type(privilege_list => xs$name_list('connect'),
                           principal_name => '<your username>',
```

```
                                    principal_type => xs_acl.ptype_db));
END;
```

5.9 Create an Oracle Enterprise Resource Planning Cloud Connection

Oracle Enterprise Resource Planning (ERP) Cloud is a cloud based end-to-end Software as a service (SaaS) suite to manage functions such as accounting, financial management, project management, procurement, and risk management. The Data Transforms Oracle ERP Cloud connector allows you to extract report data from an ERP Cloud server and load it into Autonomous Database.

To use an Oracle ERP Cloud connection in Data Transforms you need the URL of the BI Publisher web service used for the Oracle ERP Cloud instance and BI Publisher report file that is output in the Data (CSV) format.

Note the following:

- You cannot use an Oracle ERP Cloud connection as a target for mapping.
- Data load is not supported for Oracle ERP Cloud connections.

This topic has the following sections:

- Creating an Oracle ERP Cloud Server Connection
- Importing Data Entity Definitions from an Oracle ERP Cloud Server

Creating an Oracle ERP Cloud Server Connection

To create this connection, you need to specify the WSDL URL of the Report service and the username and password to connect to the ERP Cloud instance.

To define an Oracle ERP Cloud Server Connection:

1. From the left pane of the Home page, click the **Connections** tab.
   Connections page appears.
2. Click **Create Connection**.
   Create Connection page slides in.
3. Do one of the following:
   - In the **Select Type** field, enter the name or part of the name of the connection type.
   - Select the **Applications** tab.
4. Select **Oracle ERP Cloud** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. In the **WSDL URL** textbox, enter the URL of the BI Publisher web service used for the Oracle ERP Cloud instance. The format of the URL is *`BI Publisher Hostname/`*`xmlpserver/services/PublicReportWSSService?wsdl`.
   For example: `https://fa-eudu-saasfademo1.ds-fa.oraclepdemos.com/xmlpserver/services/PublicReportWSSService?wsdl`.
8. In the **Proxy Host** textbox, enter the host name of the proxy server to be used for the connection.
9. In the **Proxy Port** textbox, enter the port number of the proxy server.

10. In the **User** text box enter the user name for connecting to the Oracle ERP Cloud instance.
11. In the **Password** text box enter the password for connecting to the Oracle ERP Cloud instance.
12. After providing all the required connection details, click **Test Connection** to test the connection.
13. Click **Create**.
    The newly created connection is displayed in the **Connections** page.

    Click the **Actions** icon (⋮) next to the selected connection to perform the following operations:

    - Select **Edit** to edit the provided connection details.
    - Select **Test Connection** to test the created connection.
    - Click **Export** to export the connection. See Export Objects.
    - Select **Delete Schema** to delete schemas.
    - Select **Delete Connection** to delete the created connection.

    You can also search for the required connection to know its details based on the following filters:

    - **Name** of the connection.
    - **Technology** associated with the created connection.

Importing Data Entity Definitions from an Oracle ERP Cloud Server

For Oracle ERP Cloud connections, the Import Data Entities page requires input in key/value pairs. To import data, you need to specify the `BIPReportLocation` key, which refers to the location of the BI publisher report on the BI server. You can find this information on the BI server when the corresponding report is open.

To import data entities from Oracle ERP Cloud:

1. From the left pane of the Home page, click the **Data Entities** tab.
   Data Entities page appears.
2. Click **Import Data Entities**. The Import Data Entities page slides-in.
3. In the **Connection** drop-down, select the Oracle ERP Cloud connection that you created in Creating an Oracle ERP Cloud Server Connection. The options on the page change to display only the **Entity Name** field.
4. In the **Entity Name** field enter a name for the data entity.
5. Enter `BIPReportLocation` in the **Key** field.
6. Enter the location of the BI Publisher report file in the **Value** field corresponding to the key. Specify the path till the name of the *`ReportFile`*`.xdo` file.
7. Click **Start**.
   A Job is created, and the corresponding Job ID is displayed for you to track the session. Click the Job ID to view the details of the job. Upon successful execution of the job, the data entity is displayed on the Data Entities page.
8. Click the **Actions** icon (⋮) next to the selected Data Entity and select **Edit**.
   The **Edit Data Entity** page appears that displays a table with the following information:

    - Name

- Data Type
- Length
- Scale
- Not Null

Verify the data type and length for the imported entities. Oracle Data Transforms determines the data types and length from the first record of the file, but may set default values (for example, 50 for the string field length) or incorrect data types. In case of an empty field, data type is set to String with length 50.

9. Click **Save**.

5.10 Create an Oracle Financials Cloud Connection

You can fetch real time transactional data from Oracle Financials Cloud REST endpoints, import the data entities into Data Transforms, and use them as a source in a data flow.

To create an Oracle Financials Cloud connection you need to choose a temporary schema where Data Transforms can create data entities after the reverse-engineering operation.

To define an Oracle Financials Cloud connection,

1. From the left pane of the Home page, click the **Connections** tab.

   Connections page appears.
2. Click **Create Connection**.

   Create Connection page slides in.
3. Do one of the following:
   - In the **Select Type** field, enter the name or part of the name of the connection type.
   - Select the **Applications** tab.
4. Select **Oracle Financials Cloud** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. In the **REST Service URL** textbox, enter the URL of the endpoint that services the REST resources.
8. In the **Proxy Host** textbox, enter the host name of the proxy server to be used for the connection.
9. In the **Proxy Port** textbox, enter the port number of the proxy server.
10. In the **User** text box enter the user name for connecting to the REST endpoint.
11. In the **Password** text box enter the password for connecting to the REST endpoint.
12. Choose a connection from the **Staging Connection** drop-down list. The list displays only existing Autonomous Database connections. To use a different connection, create the connection before you reach this page.
13. After providing all the required connection details, click **Create**.

    The new connection is created.
14. Click **Test Connection**, to test the established connection.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon (⋮) next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Select **Test Connection**, to test the created connection.
- Click **Export** to export the connection. See Export Objects.
- Select **Delete Schema**, to delete schemas.
- Select **Delete Connection**, to delete the created connection.

You can also search for the required Connection to know its details based on the following filters:

- **Name** of the Connection.
- **Technology** associated with the created Connection.

5.11 Create and Use an Oracle NetSuite Connection

You can use the Oracle NetSuite JDBC Driver or OAuth 2.0 authentication to connect to the Oracle NetSuite application. For Oracle NetSuite connections, Data Transforms allows you to load pre-built dataflows and workflows that you can run to transfer data from NetSuite to your target schema.

This topic has the following sections:

- Configuring Access Permissions Required for Building Data Warehouse
- Creating the Oracle NetSuite Connection
- Using the Build Data Warehouse Wizard
- Running the Pre-Built Workflows to Load Data into the Target Schema

Configuring Access Permissions Required for Building Data Warehouse

Before you create a NetSuite connection or use the Build Data Warehouse Wizard in Data Transforms, you need to login to NetSuite as an administrator, enable SuiteAnalytics Connect, create a custom role, and set the access permissions that are required to build the NetSuite Data Warehouse.

To configure the access permissions:

1. Log in to NetSuite as an administrator using the following URL:
   https://system.netsuite.com/pages/customerlogin.jsp
2. Check whether the SUITEANALYTICS CONNECT feature is enabled.
   Go to *Setup* > *Company* > *Enable Features*. Click the **Analytics** tab and verify that the **SuiteAnalytics Connect** box is checked.
3. Add a custom role.
   Go to *Setup* > *Users/Roles* > *User Management* > *Manage Roles*. Click **New Role**, add the required details, and click **Save** to create a custom role.
4. Assign the required permissions to the custom role.
   Go to *Setup* > *Users/Roles* > *User Management* > *Manage Roles*. Click **Customize** next to the name of the custom role for which you would like to add the SuiteAnalytics Connect permission.

Click the **Transactions** tab under the **Permissions** tab and assign the following permissions:

- Account Detail
- Accounting Lists
- Accounting Management
- Accounts
- Accounts Payable
- Accounts Payable Graphing
- Accounts Payable Register
- Accounts Receivable
- Accounts Receivable Graphing
- Accounts Receivable Register
- Adjust Inventory
- Adjust Inventory Worksheet
- Amortization Reports
- Amortization Schedules
- Audit Trail
- Balance Sheet
- Bank Account Registers
- Bill Purchase Orders
- Billing Schedules
- Bills
- Bin Putaway Worksheet
- Bin Transfer
- Blanket Purchase Order
- Build Assemblies
- CRM Groups
- Calendar
- Cash Sale
- Cash Sale Refund
- Charge
- Charge - Run Rules
- Charge Rule
- Check
- Classes
- Commission Feature Setup
- Commission Reports
- Commit Orders

- Commit Payroll
- Competitors
- Component Where Used
- Contacts
- Count Inventory
- Create Allocation Schedules
- Credit Card
- Credit Card Refund
- Credit Card Registers
- Credit Memo
- Credit Returns
- Currency
- Currency Revaluation
- Custom Recognition Event Type
- Customer Deposit
- Customer Payment
- Customer Refund
- Customers
- Deferred Expense Reports
- Deleted Records
- Departments
- Deposit
- Deposit Application
- Documents and Files
- Edit Forecast
- Edit Manager Forecast
- Email Template
- Employee Commission Transaction
- Employee Commission Transaction Approval
- Employee Record
- Employee Reminders
- Employees
- Enter Opening Balances
- Enter Vendor Credits
- Equity Registers
- Establish Quotas
- Estimate
- Events

- Expense Report
- Expenses
- Export Lists
- Fair Value Dimension
- Fair Value Formula
- Fair Value Price
- Financial Statements
- Find Transaction
- Fixed Asset Registers
- Fulfill Orders
- Fulfillment Request
- General Ledger
- Generate Price Lists
- Generate Statements
- Imported Employee Expenses
- Inbound Shipment
- Income
- Income Statement
- Inventory
- Inventory Status Change
- Invoice
- Invoice Approval
- Invoice Sales Orders
- Item Fulfillment
- Item Receipt
- Item Revenue Category
- Items
- Lead Snapshot/Reminders
- Locations
- Long Term Liability Registers
- Make Journal Entry
- Manage Accounting Periods
- Manage Payroll
- Mass Updates
- Memorized Transactions
- Mobile Device Access
- Net Worth
- Non Posting Registers

- Notes Tab
- Opportunity
- Other Asset Registers
- Other Current Asset Registers
- Other Current Liability Registers
- Other Lists
- Other Names
- Ownership Transfer
- Pay Bills
- Pay Sales Tax
- Pay Tax Liability
- Paycheck Journal
- Payroll Items
- Perform Search
- Phone Calls
- Post Vendor Bill Variances
- Posting Period on Transactions
- Price Books
- Price Plans
- Process Payroll
- Project Revenue Rules
- Purchase Contract
- Purchase Order
- Purchase Order Reports
- Purchases
- Reconcile
- Reconcile Reporting
- Refund Returns
- Report Customization
- Report Scheduling
- Request For Quote
- Requisition
- Resource
- Return Authorization
- Revalue Inventory Cost
- Revenue Arrangement
- Revenue Arrangement Approval
- Revenue Element

- Revenue Recognition Field Mapping
- Revenue Recognition Plan
- Revenue Recognition Reports
- Revenue Recognition Rule
- Revenue Recognition Schedules
- SOAP Web Services
- Sales
- Sales By Partner
- Sales By Promotion
- Sales Order
- Sales Order Approval
- Sales Order Fulfillment Reports
- Sales Order Reports
- Sales Order Transaction Report
- Set Up Budgets
- Set Up SOAP Web Services
- Statement Charge
- Statistical Account Registers
- Store Pickup Fulfillment
- Subscription Change Orders
- Subscription Plan
- Subscriptions
- Subsidiaries
- SuiteAnalytics Connect
- SuiteAnalytics Workbook
- Tasks
- Tax
- Track Messages
- Transaction Detail
- Transfer Funds
- Transfer Inventory
- Transfer Order
- Transfer Order Approval
- Trial Balance
- Unbilled Receivable Registers
- Unbuild Assemblies
- Units
- Vendor Bill Approval

- Vendor Payment Approval
- Vendor Request For Quote
- Vendor Return Auth. Approval
- Vendor Return Authorization
- Vendor Returns
- Vendors
- Work Calendar
- Work Order
- Work Order Close
- Work Order Completion
- Work Order Issue

5. Add the **SuiteAnalytics Connect – Read All** permission.
   Click the **Setup** tab under the **Permissions** tab, select **SuiteAnalytics Connect – Read All** from the drop-down, and click **Add**.
6. Click **Save** to apply these permissions to the custom role.
7. Assign the custom role to a user. When you create a connection to NetSuite from Data Transforms, you will need to enter the credentials of this user to connect to the data server. See Creating the Oracle NetSuite Connection for information about creating the connection.
   To assign the custom role to the user, go to *Setup* > *Users / Roles* > *Manage Users*. Click **Edit** next to the name of the user, assign the custom role, and click **Save**.
8. To verify the access permissions, log in as the user that has the custom role assigned. Go to *Analytics* > *Datasets*. Click **New Dataset**. This page will list all the tables and record types that the user has access to. Search for “transaction” table, for example, to verify whether the user has access to the transaction table.

Creating the Oracle NetSuite Connection

You can create an Oracle NetSuite connection using JDBC connectivity or OAuth 2.0 authentication.

To define an Oracle NetSuite connection:

1. From the left pane of the Home page, click the **Connections** tab.
   Connections page appears.
2. Click **Create Connection**.
   Create Connection page slides in.
3. Do one of the following:
   - In the **Select Type** field, enter the name or part of the name of the connection type.
   - Select the **Applications** tab.
4. Select **Oracle NetSuite** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. To specify the connection details, do one of the following:
   - To use JDBC connectivity, specify the following details:

- **JDBC URL** - Enter the URL of the SuiteAnalytics Connect server to be used for the connection.
- **User** - Enter the user name for connecting to the data server.
- In the **Password** textbox enter the password for connecting to the data server.
- In the **Account ID** textbox, enter the account ID for connecting to the data server.
- In the **Role ID** textbox, enter the role ID for connecting to the data server.

- To use OAuth 2.0 authentication, click the **OAuth 2.0** switch and then specify the following details:
  - **Username** - Enter the name of the user who has role access to login to NetSuite using OAuth 2.0 connection. This is the user you have assigned the custom role to in Configuring Access Permissions Required for Building Data Warehouse.
  - **Account ID** - Enter the account ID for connecting to the data server. You can get this information by logging into the NetSuite account and viewing the SuiteAnalytics connect information.
  - **Role ID** - Enter the role ID for connecting to the data server. You can get this information by logging into the NetSuite account and viewing the SuiteAnalytics connect information.
  - **Client ID** - Enter the client ID for connecting to the data server.
    To obtain the client ID, create an Integration record in NetSuite by enabling OAuth 2.0 Client Credentials Flow. Copy and save the Client ID that is displayed when the Integration Record is successfully created.

    > **Note:**
    >
    > NetSuite no longer supports the RSA PKCSv1.5 scheme for token signing for NetSuite OAuth 2.0 client credentials flow. Any integrations that rely on the RSA PKCSv1.5 scheme will need to be updated to use the RSA-PSS scheme. Refer to the Oracle NetSuite documentation for more information.

  - **Public Certificate** and **Private Key** - Use the OpenSSL commands to generate the key pair in the required PEM format. For example,
    ```
    openssl req -new -x509 -newkey rsa:4096 -keyout private.pem -sigopt rsa_padding_mode:pss -sha256 -sigopt rsa_pss_saltlen:64 -out public.pem -nodes -days 365
    ```
    Paste the contents of `public.pem` in the Public Certificate field. Paste the contents of `private.pem` in the Private Key field.
  - **Certificate ID** - Enter the Certificate ID for connecting to the data server.
    To get the certificate ID, use the NetSuite OAuth 2.0 Client Credentials (M2M) Setup to add the public certificate file (`auth-cert.pem`) to the certificate key list and copy the generated Certificate ID.

8. If the source that you want to use for mapping is a saved search, you need to also specify the following details in **Saved Search Extraction**:
   - **Application ID**: Enter the NetSuite Application ID for Data Transforms.
   - **Version**: Enter the NetSuite version number.

9. Select the checkbox in **Build Data Model** to install pre-built dataflows and workflows that you can run to extract data from NetSuite and move it to your Oracle target schema using the Build Data Warehouse wizard.
10. Click **Test Connection**, to test the established connection.
11. After providing all the required connection details, click **Create**.
    The new connection is created.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon (⋮) next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Select **Test Connection**, to test the created connection.
- Select **Build Data Warehouse**, to select the functional areas and create the NetSuite Data Warehouse in the target schema. See Using the Build Data Warehouse Wizard for more information.
- Click **Export** to export the connection. See Export Objects.
- Select **Delete Schema**, to delete schema.
- Select **Delete Connection**, to delete the created connection.

You can also search for the required Connection to know its details based on the following filters:

- **Name** of the Connection.
- **Technology** associated with the created Connection.

Using the Build Data Warehouse Wizard

Data in your NetSuite account is grouped into business or subject areas in the Analytics Warehouse. The Build Data Warehouse wizard allows you to select the areas that you want to include in the newly created Data Warehouse.

To use the Build Data Warehouse Wizard:

1. On the Home page, click the **Connections** tab. The **Connections** page appears.
2. Click the Actions icon (⋮) next to the Oracle NetSuite connection that you want to use to build the data warehouse and click **Build Data Warehouse**.
   The Build Data Warehouse wizard opens.
3. From the **Connection** drop-down list, choose the Autonomous Database connection where your target schema resides.
4. From the **Staging Schema** drop-down, all schema corresponding to the selected connection are listed in two groups:
   - Existing Schema (ones that you've imported into Oracle Data Transforms) and
   - New Database Schema (ones that you've not yet imported).

   Select the schema that you want to use from the drop-down.
5. Similarly select the **Target Schema**.
6. Click **Next**.

7. Select the **NetSuite Business Areas** that you want to use to transfer data from the NetSuite Data Warehouse to the target schema.
8. Click **Save**.
   Data Transforms starts the process to build the data warehouse. Click **Jobs** on the left pane of the Home page to monitor the progress of the process. When the job completes successfully, Data Transforms creates a Project folder that includes all the pre-built workflows and dataflows, which you can run to transfer data from the NetSuite connection to your target schema. See Running the Pre-Built Workflows to Load Data into the Target Schema for more information.

Running the Pre-Built Workflows to Load Data into the Target Schema

When the Build Data Warehouse wizard completes successfully, Data Transforms creates a project that includes all the pre-built data flows and workflows that you can run to extract data from a NetSuite connection and load it into your target schema.

To view and run the pre-built workflows:

1. Click **Projects** on the left pane of the Home page and select the newly created NetSuite project.
2. Click **Workflows** in the left pane. The following pre-built workflows are listed in the Project Details page:
   - Stage NetSuite Source to SDS
   - Extract Transaction Primary Keys
   - Load SDS to Warehouse
   - Apply Deletes
   - All Workflows
3. Click the Actions icon (⋮) next to the workflow you want to run and click **Start**.
   Oracle recommends that you run **All Workflows** to execute all the pre-built workflows.

   To see the status of the workflow, click **Jobs** from the left pane in the current project. When the job completes successfully, all the data from the NetSuite connection is loaded into the target schema.

5.12 Create an Oracle Object Storage Connection

You can use Data Transforms to upload data from Oracle Object Storage to Autonomous Database.

The OCI Object Storage dedicated endpoints feature allows OCI customers to securely access the storage buckets. See Object Storage Dedicated Endpoints for more information. You need to use the new URL format when you create Object Storage connections in Data Transforms. For users that already have an Object Storage connection, the existing URL is automatically updated to the new URL format.

To create an Oracle Object Storage connection you need to have an Oracle Cloud Infrastructure username and an auth token. See Getting an Auth Token for information about how to generate the auth token. You need to specify these details when you define the connection in Oracle Data Transforms.

Note the following:

- To use an Oracle Object Storage connection to import data into Data Transforms, you must use a public IP address to access the compute node. If you want to use a private IP address to access the Object Storage service, make sure that you have access to the Internet.
- The supported file format for loading data from Oracle Object Storage to Autonomous Database and vice versa is CSV.
- The supported data types are Numeric, Double, String, and Date.
- Data load is not supported for Oracle Object Storage connections.
- To create a mapping to Object Storage, the source technology must support the `DBMS_CLOUD` package. Otherwise, the mapping will fail.
- Data Transforms does not support cross-realm connectivity for Object Storage. For example, you cannot connect from Data Transforms configured in a Government Cloud region to Object Storage that is in a commercial realm.

To define an Oracle Object Storage connection,

1. From the left pane of the Home page, click the **Connections** tab.

   Connections page appears.
2. Click **Create Connection**.

   Create Connection page slides in.
3. Do one of the following:
   - In the **Select Type** field, enter the name or part of the name of the connection type.
   - Select the **Databases** tab.
4. Select **Oracle Object Storage** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. Enter the URL in the **Object Storage URL** textbox. You can enter the URL in either of the following formats:
   - URL with fully qualified domain name.
     For example,

     ```
     https://<namespace>.swiftobjectstorage.<your-region>.oci.customer-oci.com/v1/<your-namespace>/<your-bucket>
     ```

     ```
     https://<namespace>.objectstorage.<your-region>.oci.customer-oci.com/n/<your-namespace>/b/<your-bucket>/o
     ```
   - If you want to use the URL provided by the OCI Console, specify the URL only till the name of the bucket.
     For example,

     ```
     https://<namespace>.swiftobjectstorage.<your-region>.oci.customer-oci.com/v1/<your-namespace>/<your-bucket>
     ```

     ```
     https://<namespace>.objectstorage.<your-region>.oci.customer-oci.com/n/<your-namespace>/b/<your-bucket>/o
     ```

- If you choose **Credential** as the **Connection Mode** (see step 6), specify the URL till *bucketname*/o
  For example,

  ```
  https://<namespace>.objectstorage.<your-region>.oci.customer-oci.com/n/
  <your-namespace>/b/<your-bucket>/o/
  ```

  > **Note:**
  >
  > Credential mode is available only to Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

  The values for Region, Namespace and Bucket are auto-populated based on the URL provided.

8. To select the **Connection Mode** do one of the following:
   - Select **Swift Connectivity**, and provide the following details:
     - In the **User Name** text box enter your Oracle Cloud Infrastructure username. For tenancies that support identity domains, specify the domain name along with the username. For example, *<identity-domain-name>/<username>*.
     - In the **Token** text box enter the auth token.
   - (This is applicable only to Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.) Select **Credential** and provide the ODI credential in the **Enter Credential** text box.
     You must create the credential in the repository and in the Autonomous Database that you created during instance creation. When you create a data flow to map data from Object Storage to Autonomous Database you need to create the ODI credential in the target schema as well. Before you run the mapping, make sure that you select the step and in the Properties panel, set the **Create credential** KM option to `false`. Otherwise, the credential-based connection will fail.

     To create the credential, execute the following script:

     ```
     begin DBMS_CLOUD.create_credential( credential_name => '<Credential
     name>',
           username => '<oci username>', password => '<auth token>' ); end;
     ```

9. Click **Create**.

   The new connection is created.

10. Click **Test Connection**, to test the established connection.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon (⋮) next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Select **Test Connection**, to test the created connection.
- Click **Export** to export the connection. See Export Objects.
- Select **Delete Schema**, to delete schemas.

- Select **Delete Connection**, to delete the created connection.

You can also search for the required Connection to know its details based on the following filters:

- **Name** of the Connection.
- **Technology** associated with the created Connection.

5.13 Create a REST Server Connection

You can connect to any REST service endpoint, import the data entities into Data Transforms, and use them as source in a data flow.

To create a generic REST connector, you need to provide the JDBC URL, username, and password to connect to the endpoint. You can also create and upload a config file that contains information such as the authentication methods, endpoints, and tables that you want to import data entities from.

The **Application** tab on the **Create Connection** page includes two connection options to create a generic REST connection - Generic REST and Generic REST Config. This topic has the following sections:

- Creating a Generic REST Connection
- Creating a Generic Rest Connection Using a Config File

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon (⋮) next to the selected connection to perform the following operations:

- Select **Edit** to edit the provided connection details.
- Select **Test Connection** to test the created connection.
- Click **Export** to export the connection. See Export Objects.
- Select **Delete Schema** to delete schemas.
- Select **Delete Connection** to delete the created connection.

You can also search for the required connection to know its details based on the following filters:

- **Name** of the connection.
- **Technology** associated with the created connection.

Creating a Generic REST Connection

To create this connection you need to specify the REST service URL and choose a temporary schema where Data Transforms can create data entities after the reverse-engineering operation.

To define a REST server connection:

1. From the left pane of the Home page, click the **Connections** tab.
   Connections page appears.
2. Click **Create Connection**.
   Create Connection page slides in.
3. Do one of the following:

- In the **Select Type** field, enter the name or part of the name of the connection type.
- Select the **Applications** tab.

4. Select **Generic Rest** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. In the **REST Service URL** textbox, enter the URL of the endpoint that services the REST resources.
8. In the **Proxy Host** textbox, enter the host name of the proxy server to be used for the connection.
9. In the **Proxy Port** textbox, enter the port number of the proxy server.
10. In the **User** text box enter the user name for connecting to the REST endpoint.
11. In the **Password** text box enter the password for connecting to the REST endpoint.
12. Choose a connection from the **Staging Connection** drop-down list. The list displays only existing Autonomous Database connections. To use a different connection, create the connection before you reach this page.
13. After providing all the required connection details, click **Test Connection** to test the connection.
14. Click **Create**.
    The new connection is created.

Creating a Generic Rest Connection Using a Config File

APPLIES TO: Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

To create a generic REST connector, you need the JDBC URL, username, password, and a config file. The config file is a model file with the *file_name.rest* naming convention that you need to upload when you create a REST Server connection. You need to specify the endpoints, table mappings, and the authentication methods to create the config file. You can create the config file using any text editor.

To define a REST server connection using a config file:

1. From the left pane of the Home page, click the **Connections** tab.
   Connections page appears.
2. Click **Create Connection**.
   Create Connection page slides in.
3. Do one of the following:
   - In the **Select Type** field, enter the name or part of the name of the connection type.
   - Select the **Applications** tab.
4. Select **Generic Rest Config** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. Use the **Config File** text box to upload the config file that you want to use.
8. In the **JDBC URL** textbox, enter the URL to connect to the server.

9. In the **User** and **Password** text boxes enter the user name and password for connecting to the REST endpoint. You may leave these fields blank if these values are not applicable or are already mentioned in the JDBC URL.
10. After providing all the required connection details, click **Test Connection** to test the connection.
11. Click **Create**.
    The new connection is created.

6
Data Loads

A data load allows you to load multiple data entities from a source connection to a target connection.

Topics

- Create a Data Load
  Use the Create Data Load wizard to define the source connection and the target connection for the data load.
- Run a Data Load
  After you create the data load, you are taken to the Data Load Detail page that displays the details that you need to run a data load.
- Monitor Status of Data Loads, Data Flows, and Workflows
  When you run a data load, data flow, or workflow Oracle Data Transforms runs jobs in the background to complete the request. You can view the status of the job in the panel on the bottom right of the Data Load Details, the Data Flow Editor, and the Workflow Editor page.

6.1 Create a Data Load

Use the Create Data Load wizard to define the source connection and the target connection for the data load.

> **Note:**
>
> Data load is not supported for Oracle Object Storage connections.

The wizard also allows you to choose the load processing option based on the deployment type. The newly created data load is displayed in the Data Loads page of the associated project. Click the Actions icon (⋮) next to the selected data load to edit, rename, start, export, or delete it.

To create a data load from the Home page:

1. Do one of the following:
   - On the Home page, click **Load Data**. The Create Data Load wizard appears.
     In the Create Data Load tab, enter a name if you want to replace the default value, add a description, and select a project from the drop-down.
   - On the Home page, click **Projects**, and then the required project tile. In the left pane, click **Data Loads**, and then click **Create Data Load**.

   The Create Data Load wizard appears.
2. In the Name field, enter a name for the data load. The field is pre-populated with a default name. You can edit this value.
3. Add a description. This is optional.

4. If you have opened the Create Data Load wizard from the Home page, then you need to select a project name from the drop-down. If this your first time here, click the + icon to create a project. If you have logged in as SUPERVISOR, the default project name is `Home`. For other users, the default project name is in the format `<username>_Home`. You can edit the default value. See Work with Projects for more information about projects.
5. For **Load Processing** do one of the following:
   - Select the **Internal** radio button and from the **Deployment Type** drop-down select **Data Transforms (Batch)**.
   - Select the **Delegate** radio button and from the **Deployment Type** drop-down select OCI GoldenGate. From the GoldenGate Deployment Connection drop down select a connection.
6. Click **Next**.
7. To define your source connection, from the **Connection** drop-down, select the required connection from which you wish to add the data entities. Alternatively, click the + icon to create a new connection. See Work with Connections for more details about connections.
8. In the **Schema** drop-down, all schema corresponding to the selected connection are listed in two groups:
   - Existing Schema (ones that you've imported into Oracle Data Transforms) and
   - New Database Schema (ones that you've not yet imported).

   Select the schema that you want to use from the drop down.

   > **Note:**
   >
   > If there is missing information such as user name or password not specified, wallet missing, and so on, the list may fail to populate with a “This connection has missing information.” error. Click the Edit icon ( ) to open the Update Connection page where you can fill in the missing details.

9. Click **Next**.
10. Similarly, define the target connection. If you plan to use an Apache Iceberg connection as a target to load data, see Creating and Running an Apache Iceberg Data Load for specific instructions.
11. Click **Save**.
    The Data Load Detail page appears listing all the loaded data entities.

6.2 Run a Data Load

After you create the data load, you are taken to the Data Load Detail page that displays the details that you need to run a data load.

It includes the details of the source schema, the data entities that are loaded from the source schema, and the details of the target schema. You can choose the action that you want to apply on each data entity – recreate, truncate, append - and load tables in bulk to the target schema. When you run a data load, multiple jobs run in the background to complete the request.

You can run a data load in either of the following ways:

- Using the Data Load Detail Page.

- Running a Data Load from the Workflow Details Page.

> **Note:**
>
> Data load is not supported for Oracle Object Storage connections.

Running a Data Load from the Data Load Detail Page

The Data Load Detail page displays the information that you need to run a data load. You can apply different actions - incremental merge, incremental append, recreate, truncate, append - on the data entities before loading it to the target schema.

> **Note:**
>
> APPLIES TO: Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.
> If the data load is huge, you might want to increase the memory of the ODI Agent to avoid any issues. Follow the instructions in Increase the Memory of ODI Agent before you start to run the data load.

To run a data load from the Data Load Detail Page:

1. In the Data Load Detail page, select the data entities that you want to move to the target schema.
   To filter the list, you can do one of the following:
   - Enter the name or part of the name in the **Name** text box. This search returns data entities that include the specified string in the name. For example, the search string AD returns results such as ADMIN, ADDRESS, BADGE, UPGRADE, WORKLOAD, and so on.
   - Turn on the **Use Regular Expression** toggle to use pattern matching to search for particular strings of characters. For example, the search string CO.* returns results such as CONTACT, COUNT, COUNTRY and so on.

   You can select multiple data entities and load them to the target schema. You can also sort the displayed list using the following options:
   - All - Displays all the data entities that match the search criteria.
   - Selected - Displays all the rows that you selected.
   - Unselected - Displays all the unselected rows.
   - Invalid – Displays all the invalid rows.

   > **Note:**
   >
   > These options display the list of data entities based on the search criteria. To view the list of all data entities, clear any applied filters.

2. Click on the required icon to choose any of the following actions:
   - Incremental Merge - Updates the data in the selected column by comparing the source table with the target table based on the specified merge key. To use this option, select

the column that you want to merge and then select the merge key. To use this option, select the column that you want to merge and then select the merge key. Click the Validate icon ( ) to validate the selected values.

> **Note:**
>
> – The last update values for the Incremental Column are stored in the Data Load. If you create a new data load with the same table, this option acts like Append, Truncate, and Recreate the first time you run it.
>
> – If the target table is not truncated, do not start the Data Load using the Append, Truncate, or Recreate mode and then switch the action to Incremental Merge. This will cause every existing row in the target table to be updated. Oracle recommends that you start the Data Load using the Incremental Merge action.

- Incremental Append - Updates data in the selected column in the target schema. To use this option, select the column that you want to update and click the Validate icon ( ) to validate the selection.
  The first time you run the Data Load, this option works in the same way as the Append, Truncate, and Recreate options, meaning all rows are inserted into the target. On all subsequent Data Load runs, the Incremental Column is used to filter only new rows to append data to avoid the same rows from being loaded more than once.

  > **Note:**
  >
  > – The last update values for the Incremental Column are stored in the Data Load. If you create a new data load with the same table, this option acts like Append, Truncate, and Recreate the first time you run it.
  >
  > – If the target table is not truncated, do not start the Data Load using the Append, Truncate, or Recreate mode and then switch the action to Incremental Append. Oracle recommends that you start the Data Load using the Incremental Append action.

- Recreate – If the table is already present in the target schema, drops the existing table and recreates it.

  > **Note:**
  >
  > This option is not available for data entities that are loaded using OCI GoldenGate.

- Truncate – If the table is already present in the target schema, deletes all the data from the selected table. Nothing is dropped.

> **Note:**
>
> For Delta Share data loads the Data Load Detail page only includes the option. You cannot apply different actions - incremental merge, incremental append, recreate, truncate, append - on the data entities before loading it to the target schema. This is to make sure that the data is consistent between the Delta Sharing server and the target schema.

- Append – Inserts all rows in the Dataset into the target. If the Dataset contains records that already exist in the target, there will be duplicate records. Use Append if the source dataset is expected to have only new records or duplicate rows in the target are not a concern.
- Do Not Load – Skips the selected data entity from the data load job. After you click **Save**, these data entities are no longer available for future data load jobs.

You can select multiple data entities and apply different actions. The unsaved rows are highlighted in bold.

> **Note:**
>
> These options are not available for Delta Share connections.

3. To specify how you want to store the source column names in the target tables, click **Settings**, which is on the right side of the Data Load Detail page. Choose one of the following:
   - **Retain original names by enclosing all names with delimiters** - Creates column names with the same names as is from the source tables in the target table.
   - **Use no delimiters** - This is the default selection. Converts all the column names to upper case and replaces spaces and special characters with underscores.
     The following options are applicable to reserved words such as `Date`, `Timestamp`, `Start`, and so on.
     - **Enclose with delimiters** - This is the default selection. Encloses column names that are reserved words with delimiters (not all column names).
     - **Use a prefix** - Adds the specified prefix to column names that are reserved words (not all column names).

     For column names that have the same name after conversion, the names are suffixed with a numeric value to maintain uniqueness. For example column names

     `Date, date, DATE, Item_@Code, Item$$Code, Item%%Code`

     are created in the target table as

     `DATE, DATE_0, DATE_1, ITEM__CODE, ITEM__CODE_0, ITEM__CODE_1.`

> **Note:**
>
> Once the data load is run, the selected options are applied and retained for all subsequent runs. You cannot change the configuration.

4. Click  to save the changes. A green checkmark () in the row indicates that the changes are saved.
5. To start the data load,
   - Click  .
   - For GoldenGate data loads, click  .

   A confirmation prompt appears when the data load starts successfully.

To check the status of the data load, see the Status panel on the right below the Target Schema details. For details about the Status panel, see Monitor Status of Data Loads, Data Flows, and Workflows. This panel shows links to the jobs that execute to run this data load. Click the link to monitor the progress on the Job Details page. For more information about jobs, see Create and Manage Jobs.

Running a Data Load from the Workflow Details Page

You can add multiple data loads to a workflow along with data flows or workflows and run them as separate steps. The left panel of the Workflow Details page lists the data flows, workflows, and data loads that are available for use.

For data loads, the left panel lists the following two folders:

- **Data Loads** - This folder lists all the data loads that you have created in the local Data Transforms instance. When you select a step in the workflow that is a local data load, the Properties Panel available on the right side of the design canvas displays the Type as `Data Load` and the Linked Object as Home>`<nameofDataLoad>`.
- **Remote Data Loads** - This folder lists all the data loads that you have created in a remote Data Transforms instance. See Create a Data Transforms Connection for Remote Data Load. When you select a step in the workflow that is a remote data load, the Properties Panel available on the right side of the design canvas displays the Type as `Remote Data Load` and the Linked Object as `Name of the Data Transforms connection` > `<nameofDataLoad>`.

After you have added the data loads to the workflow, click  to execute them.

6.3 Monitor Status of Data Loads, Data Flows, and Workflows

When you run a data load, data flow, or workflow Oracle Data Transforms runs jobs in the background to complete the request. You can view the status of the job in the panel on the bottom right of the Data Load Details, the Data Flow Editor, and the Workflow Editor page.

This panel includes the following:

- A Refresh icon () to refresh the displayed status.
- The current status. You can see any of the following statuses:
  - Not Started – If no data load, data flow, or workflow has been executed yet.
  - Running – When the job starts.
  - Done – When the job completes.
    The status of the last job run persists in this panel till you run a new data load, data flow, or workflow.

> **Note:**
>
> For data loads created using OCI GoldenGate, the status panel shows the link to the GoldenGate Deployment Console and the status of the Extract and Replicat processes.

- The name of the job and a link that takes you to the Jobs Details page where you can see detailed information about the job and monitor the execution status. For more information, see Create and Manage Jobs.
  When you run a data load, multiple jobs run in the background to complete the request. This panel shows a link for each job that executes when running a data load.

7

Data Entities

Learn about how you can create and import data entities that you can then use to create data flows to transform data.

Topics

7.1 Work with Data Entities

A Data Entity is a tabular representation of a data structure.

It includes Database Tables or Views that can be used in a mapping as a source or target. They are simply the metadata for sources and targets. They are used in creating data flows.

You can add Data Entities to your newly created project in one of the following two ways:

- Importing Data Entities
- Creating Data Entities

All the newly created or imported Data Entities along with their details are displayed in the **Data Entities** page. The details include:

- **Name** of the Data Entity
- **Connection** for which the Data Entity was created
- **Schema** to which the Data Entity is associated

- Click the **Actions** icon (⋮) next to the selected Data Entity to perform the following operations:
  - Select **Edit**, to edit the existing details.
  - Select **Preview**, to preview the selected Data Entity. If the data entity belongs to an Oracle database, you can also view statistics of the table. See View Statistics of Data Entities for more details.

  - Select **Delete**, to delete the selected Data Entity.
- To delete the Data Entities in bulk, in the Data Entities page, select the check boxes of the respective Data Entities and click **Delete**.
- You can also search for the required Data Entity to know its details based on the following filters:
  - **Name** of the Data Entity
  - **Connection** for which the Data Entity was created
  - **Schema** to which the Data Entity is associated
  - **Tag** that is associated with the Data Entity

7.2 Import Data Entities

The easiest and most common way to create a Data Entity is by importing its structure from the connection type (for example, Oracle database).

To import existing Data Entities:

1. From the left pane of the Home page, click the **Data Entities** tab.
   Data Entities page appears.
2. Click **Import Data Entities**, to import the existing data entities.
   Import Data Entities page slides-in. If you plan to use an Oracle ERP Cloud Server connection for the import, see Importing Data Entity Definitions from an Oracle ERP Cloud Server.
3. Select the **Connection** followed by **Schema** and then finally select the **Type of Objects** you want to import.
   For Oracle Object Storage connections, the Schema drop-down lists the name of the bucket that you specified in the URL when you created the connection.
4. [For Oracle Business Intelligence Cloud Connector (BICC) connections only] From the **Offerings to import for collection**, choose the offerings whose data stores you want to import. You must select at least one offering to import the BICC public view objects (PVO).

   > **Note:**
   >
   > The import of BICC PVOs can take a long time depending on the number of selected objects. To improve performance, Oracle recommends that you use a mask to filter and limit the number of PVOs that you want to import.

5. Choose a **Mask**/filter if you don't want to import every object in the schema.
   Depending on the Connection Type, you will be presented with further options for importing.

   > **Note:**
   >
   > For Oracle Object Storage connections, this value is case-sensitive. If **Batch similar files** is set to `True`, all the files that match the mask and have the same structure are grouped together into a single data entity.

6. [For Oracle Financials Cloud connections only] From the list in the **Resources** section, select the items that you want to import. When the import process completes, a table is created for each selected resource.
7. [For REST server connections only] In the **Resources** section, do the following:
   - In the **Resource URI** field, enter the URL of the REST service you want to import resources from.
   - Click the + icon.
   - In the **Name** column enter an identifier for the resource.
   - In the **Operation URI** column enter the URI of the resource.
   - Click **Test Resource** to check whether the entries are valid.
8. Click **Start**.
   A Job is created and the corresponding Job ID is displayed for you to track the session. Click the Job ID to view the details of the job.

   Upon successful execution of the job, all the selected Data Entities are imported. Click the Refresh icon present at the right corner of the Data Entities page, to see the new imported Data Entities.

7.3 Create Data Entities

You can manually create or update the Data Entities through the Oracle Data Transforms interface.

Data entities should possess the corresponding objects in the source connection to be used as a source in a data flow. Usually the import process makes sure that these objects are in coordination. However, whenever you manually create or update Data Entities always make sure to check if both the definitions are in coordination with each other.

When you use a Data Entity as a target then it doesn't have to exist previously in the target connection and can be created as a part of Data Flow execution.

To create a new Data Entity:

1. From the left pane of the Home page, click the **Data Entities** tab.
   Data Entities page appears.
2. Click **Create Data Entity**, to create a new data entity.
   Create Data Entity page appears.
3. In the Name text box, enter the name of the new Data Entity that you are creating.
4. From the **Connection** drop-down, select the required connection from which you wish to add the newly created Data Entity.

   > **Note:**
   >
   > Oracle Financials Cloud connections are not listed here because you cannot manually create data entities for such connections. You can only import data entities from Oracle Financials Cloud REST endpoints using the Import Data Entities page. See Import Data Entities.

5. In the **Schema** drop-down, all schema corresponding to the selected connection are listed in two groups.

- New Database Schema (ones that you've not imported from before) and
- Existing Database Schema (ones that you've imported from before and are potentially replacing data entities).

From the **Schema** drop-down, select the required schema.

> **Note:**
>
> For Oracle Object Storage connections, the Schema drop-down lists the name of the bucket that you specified in the URL when you created the connection.

6. From the **Type** drop-down, select the data entity type.
   - **Table**: To define the table structure for the newly created Data Entity, click the + icon to add columns. For each column, you can specify parameters such as Name, Data Type, Length, Scale, Not Null. Double click on the cell to configure the value.
     Click the 'x' icon, to delete a row. Click the Up and Down arrows to sort the table rows.
   - **Inline View**: To create the data entity using inline code, enter the Select statement in the **Query** tab. For example, `SELECT * FROM CUSTOMER`. Click **Validate**.
     The **Columns** tab displays a read-only list of the columns that the query returns. Click the **Preview** tab to see the column data.
7. In the **Tags** text box, enter a tag of your choice. You can use tags to filter the Data Entities displayed in the Data Entity Page.
8. For Oracle Object Storage connections, this page displays the following options:
   - **Contents** – Select the CSV file that contains the data you want to import. The metadata displayed in the preview table, such as the data type and length of columns, is based on the first row of the CSV file. Make sure that the CSV file has a header line. The header should contain only alphanumeric characters and no special characters.
   - **Group Files** – Select this check box if you want to group data from multiple CSV files into one data entity. For example, say you want to merge data from Employee_Data1.csv, Employee_Data2.csv, and Employee_Data3.csv into a single data entity.
     If you want to upload the CSV files using Windows, make sure that you first convert the files to Unix format to avoid character encoding issues during grouping.
   - **Resource Name** – Use this option along with **Group Files**. Enter the value you want to use to identify the files. The resource name should be a regular expression. You can use only an asterisk (*) as a wildcard character in the resource name. For example, `Employee_Data*.csv`.
9. For Oracle database connections, you can mark the data entities as a feature group. Expand **Advanced Options** and click the **Treat as Feature Group** checkbox.
10. Click **Save**.
    The new Data Entity is created.

7.4 Create Data Entities within the Data Flow editor

If you have already created or imported your target data entity, then you would drag the data entity onto the Design Canvas and complete the column mappings and options.

To create the definition of an entity while in the Data Flow editor,

1. Drag the data entity onto the Design Canvas.

2. Select the component and click the Add Data Entity icon present on the top right corner of the target component.
3. **Add Data Entity** page appears allowing you to configure the following details of the target component:
General tab

- In the **Name** text box, enter the name of the newly created Data Entity.
- From the **Entity Type** drop-down, select the data entity type.
  - **Table**
  - **ML Model**
    When you select this entity type the user interface changes as follows:
    * The **Connection Type** drop-down only lists **Oracle** as the option. Consequently, the **Connection** drop down only lists the Oracle connections that you have created.
    * The Add Data Entity wizard displays the **Properties** tab where you can select the Type of Learning, Function, Algorithm, and configure parameters to define the ML Model. See Machine Learning (ML) Models for detailed information about creating and using an ML Model data entity.
- From the **Connection Type** drop-down, select the required connection from which you wish to add the newly created Data Entity. The Connection drop-down is populated with the connections you have created with the associated connection type.
- From the **Connection** drop-down, select the server name from which you wish to add the newly created Data Entity.
- In the **Schema** drop-down, all schema corresponding to the selected connection are listed in two groups.
  - New Database Schema (ones that you've not imported from before) and
  - Existing Database Schema (ones that you've imported from before and are potentially replacing data entities).

  From the **Schema** drop-down, select the required schema.

  > **Note:**
  >
  > For Oracle Object Storage connections, the Schema drop-down lists the name of the bucket that you specified in the URL when you created the connection.

- In the **Tags** text box, enter a tag of your choice. You can use tags to filter the Data Entities displayed in the Data Entity Page.
- For Oracle database connections, you can mark the data entities as a feature group. Expand **Advanced Options** and click the **Treat as Feature Group** checkbox.
- Click **Next**.

Columns tab

- Click the  Add Columns icon, to add new columns to the newly created Data Entity.
  A new column is added to the displayed table.

- The table displays the following columns:
  - Name
  - Data Type - Click the cell to configure the required Data Type.
  - Scale
  - Length
  - Actions - Click the cross icon to delete the created column.
- To delete the columns in bulk, select the columns and click the  Delete icon.
- To search for the required column details, in the Search text box enter the required column name and click enter. The details of the required column are displayed.
- Click **Next**.

Preview Data Entity tab

It displays a preview of all the created columns and their configured details. If the data entity belongs to an Oracle database, you can also view statistics of the table. See View Statistics of Data Entities for more information.

4. Click **Save**.
   The new target Data Entity is created.
5. Expand the **Properties Panel** in the right pane to view the following settings of the created components:
   - General - Displays the Name of the component along with its Connection and Schema details.
   - Attributes - Displays the details of all the attributes associated to the component.
   - Column Mapping - Click Auto Map to map all the columns automatically.
   - Preview - Click to have a preview of the component.
   - Options - Change the options as appropriate.

7.5 View Statistics of Data Entities

The Preview tab displays detailed statistics of each data entity.

> **Note:**
>
> This feature is available for Oracle database tables only.

You can view the statistics of a selected data entity in one of following ways:

- In the Data Entities list, click the **Actions** icon (⋮) next to the Data Entity and click **Preview**. Select the **Statistics** tab to view the statistics of the selected data entity.
- On any data flow click on any source or target data entity, and expand the properties panel in the right pane. Click **Preview**.

The statistical data is presented as follows:

- The total number of rows and columns in the data entity is displayed at the top.

- The statistics panel displays the thumbnail graphs for each column with information about the Min, Max, Distinct, and Null values.
- Two types of thumbnail representations are displayed based on the histogram:
  - A bar chart represents data for frequency and top-frequency histograms. The bar chart show the first top 10 values for the number of rows in the table.
  - A table lists data for Hybrid and Height-Balanced histograms. The table displays the entire data and is scrollable. The table displays the range for the values and the percentage of rows in each range.
- You can click each thumbnail to view the statistics of the column in a new browser tab.
- The detailed view of each chart also shows the type of histogram.

8
Projects

Know how to manage and work with projects.

Topics

- Work with Projects
  A project is the top-level container, which can include multiple folders to organize your data flows or work flows into logical groups.

8.1 Work with Projects

A project is the top-level container, which can include multiple folders to organize your data flows or work flows into logical groups.

The Projects page displays all the existing projects, which includes projects created by all users not just the logged in user.

You can perform the following operations on a Project folder:

- View Details
- Edit the name
- Export the Project. See Export Objects.
- Delete

Creating a Project

To create a new project:

1. Click **Create Project** on the Projects page.
2. On the **Create Project** page, provide a project name.
3. Click **Create**.
   After creating the project, you're redirected to the **Projects** page.

Viewing Project Details

On the Projects page, select a project and click **View Details** from the project's **Actions** icon (⋮) or select a project to open its details page. The left panel displays all the resources associated with the selected project such as data loads, data flows, workflows, variables, schedules, and jobs.

Deleting a Project

When you delete a project, you also delete the resources it contains. Once you delete a project, it cannot be restored.

> **Note:**
>
> Be sure to review all contained resources before you delete the project.

To delete a project, on the Projects page, select Delete from the **Actions** icon (⋮) for the project you want to delete or select a project to open its details page, and then click **Delete**.

In the **Delete Project** dialog, click **Delete** to confirm the delete operation.

9
Data Flows

Get detailed information about creating and working with data flows, which includes adding components, mapping columns, and executing a data flow.

Topics

9.1 About Data Flows

A data flow defines how the data is moved and transformed between different systems.

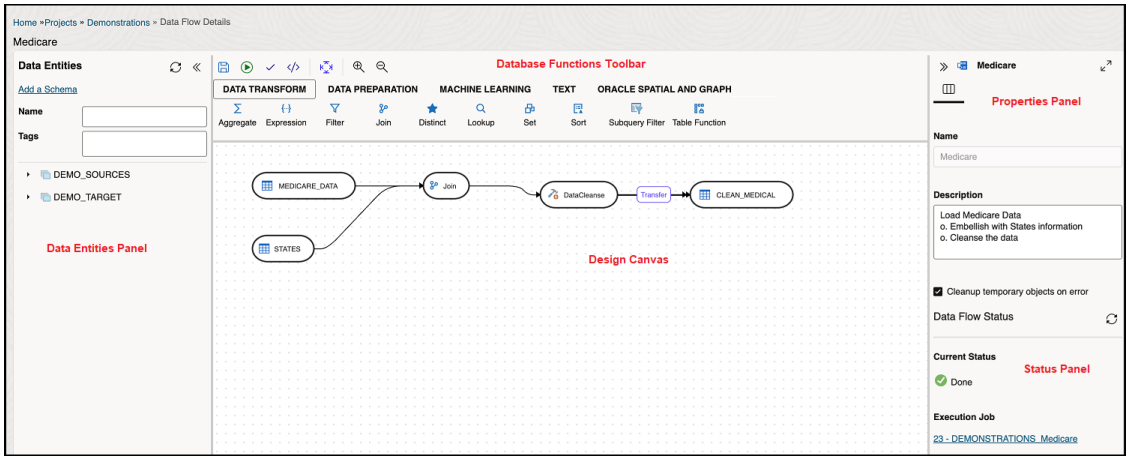
A data flow in Data Transforms connects sources to targets through a flow of components such as Join, Filter, Aggregate, Set, Split, and so on. See Supported Database Functions for more information.

When you run a data flow, Data Transforms uses the joins, filters, mappings, and constraints to transform source data and load it to target tables. Note that you can run only one execution flow at a time. You cannot put multiple flows on a Data Flow and a flow cannot diverge into multiple flows.

Topics

9.2 About Data Flow Editor

The Data flow editor is divided into five parts, the Data Entity Panel, the Database Functions Toolbar, the Design Canvas, the Properties Panel, and the Status Panel.



- **Data Entities Panel**: The data entity panel displays the Data Entities that are available to use in your Data flows. The displayed list can be filtered using the Name and Tags fields. The panel includes options that let you add schemas, import data entities, remove any of the schemas that are associated with the data flow, and refresh data entities. See Add Components for information about how to use these options.
- **Database Functions Toolbar**: The Database Functions toolbar display the database functions that can be used in your data flows. Just like Data Entities, you can drag and drop the Database tools you want to use on the design canvas. See Supported Database Functions for more information.
- **Design Canvas**: The design canvas is where you build your transformation logic. After adding the Data Entities and Database Functions to the design canvas, you can connect them in a logical order to complete your data flows.
- **Properties Panel**: The properties panel displays the properties of the selected object on the design canvas. The Properties Panel is grouped into four Tabs. **General**, **Attributes**, **Preview Data**, **Column Mapping**, and **Options**. Not all tabs are available as they vary based on the selected object. See Component Properties to know more about these options.
- **Status Panel**: When you run a data flow, the Status Panel shows the status of the job that is running in the background to complete the request. You can see the status of the job that is currently running or the status of the last job. For more information about the Status panel, see Monitor Status of Data Loads, Data Flows, and Workflows.

After designing the required data flow,

- Click , to save the created/designed data flow.

- Click , to align the nodes of the designed data flow.
- Click , to execute the created data flow.
- Click , to validate the created data flow.
- Click , , to maximize or minimize the created data flow diagram in the design canvas.

9.3 Create a Data Flow

Create data flows to load data from a source connection, run transformations, and move the data to a target database.

You can create data flows in any of the following ways:

- From the Projects page
- From the Data Flows page within a project
- From the Home page

The newly created data flow is displayed in the Data Flows page of the associated project.

Click the Actions icon ( ) next to the selected data flow to edit, rename, copy, change folder, start, export, or delete it.

From the Projects page

To create a data flow from the Projects page,

1. On the Projects page, click **Create Data Flow**.
   Create Data Flow page appears:
2. In the **Name** field, enter a name for the new data flow.
3. Select **Create New Project**, if you wish to create new project folder for the newly created data flow.
4. Else, click **Add to Existing Projects**, if you wish to add the newly created data flow to an existing project folder.
5. If you have selected **Create New Project** for the previous option, in the **Project Name** field, enter the name of the newly created project.
6. Else, if you have selected **Add to Existing Projects** for the previous option, select the required project from the **Project Name** drop-down arrow.
7. In the **Description** field, enter a description for the newly created data flow.
8. Click **Create**.

From the Data Flows page within a project

To create a data flow from the Data Flows page within a project,

1. On the Projects page click the project tile you wish to create a new data flow for. The **Project Details** page appears.
2. In the **Data Flows** page, click **Create Data Flow**.
3. Provide the **Name** and **Description** of the new data flow.

4. Click **Next**.
5. To define your source connection, from the **Connection** drop-down, select the required connection from which you wish to add the data entities.
6. In the Schema drop-down, all schema corresponding to the selected connection are listed in two groups:
   - Existing Schema (ones that you've imported into Oracle Data Transforms) and
   - New Database Schema (ones that you've not yet imported).

   Select the schema that you want to use from the drop down. For Oracle Object Storage connections, the Schema drop-down lists the name of the bucket that you specified in the URL when you created the connection.
7. Click Save.
   The Data Flow Editor appears that allows you to create a new data flow.

From the Home page

To create a data flow from the Home page,

1. On the Home page, click **Transform Data**. The Create Data Flow page appears.
2. Provide the **Name** and **Description** of the new data flow.
3. Select a project name from the drop-down. Alternatively, click the + icon to create a project.
4. Click **Next**.
5. From the **Connection** drop-down, select the required connection from which you wish to add the data entities. Alternatively, click the + icon to create a new connection.
6. In the Schema drop-down, all schema corresponding to the selected connection are listed in two groups:
   - Existing Schema (ones that you've imported into Oracle Data Transforms) and
   - New Database Schema (ones that you've not yet imported).

   Select the schema that you want to use from the drop down.
7. Click **Save**.

9.4 Add Components

Add the data entities and database functions to the Design Canvas, and connect them in a logical order to complete your data flows.

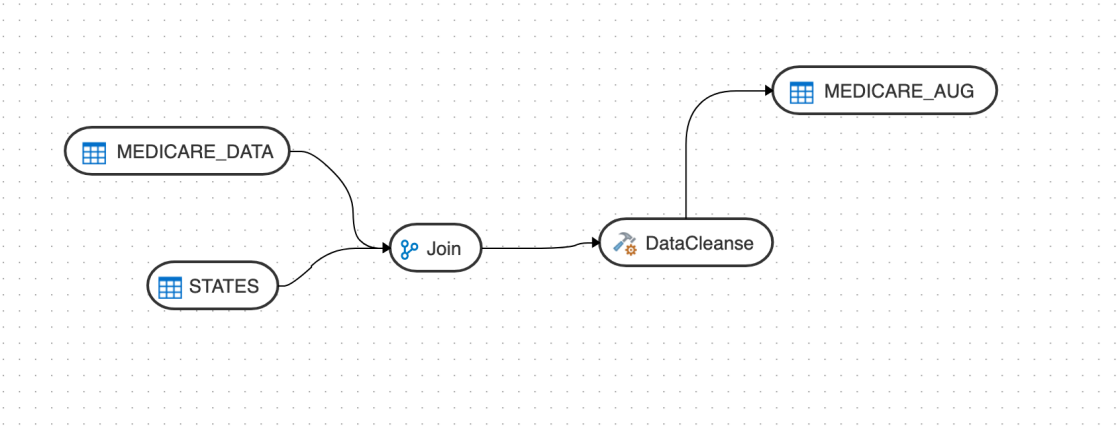
To add components to your data flow:

1. In the Data Entities panel, click **Add a Schema** to add schemas that contain the data entities that you want to use in the data flow.
2. In the Add a Schema page, select the connection and schema name.
3. Click **Import**.
4. In the Import Data Entities page, select the Type of Objects you want to import. Choose a Mask/filter if you don't want to import every object in the schema and click **Start**.
5. The Data Entities panel lists the imported data entities. The panel includes various options that let you do the following:
   - Refresh Data Entities – Click the Refresh icon to refresh the displayed list.

- **Name** - Search for data entities by name.
- **Tags** - Filter the data entities by the name of the tag used.
- **Import Data Entities** - Right-click the schema to see this option. Use this option to import the data entities.
- **Remove Schema** - Right-click the data entity to see this option. Use this option to remove the schema from the list. Note that this option does not delete the schema, it only removes the association of the schema with this data flow.

6. Similarly add more schemas to the Data Flow, if required.
7. Drag the required Data Entities that you want to use in the data flow and drop them on the design canvas.
8. From the **Database Functions** toolbar, drag the transformation component that you want to use in the data flow and drop them on the design canvas. You can use variables in the data flow. See Use Variables in a Data Flow for more information.
9. Select an object on the design canvas, and drag the **Connector** icon ( ) next to it to connect the components.
10. After saving the data flow, there may be a Transfer icon overlaid on one or more of the component connections. This indicates that ODI has detected an additional step and it is required to move the data between data servers. You can click on this Icon to view properties associated with this step.

For example:



9.5 Use Text Embedding Vector in a Data Flow

Data Transforms supports the use of vector datatype and embedding vectors in a data flow. Currently, Data Transforms integrates with OCI Generative AI service to convert input text into vector embeddings that you can use for data analysis and searches.

For text embedding Data Transforms supports both the text stored in a column and from the http links stored in a column. Before you use embedding vectors in a data flow, you need to do the following:

- Create an Oracle Database 23ai connection. See Work with Connections for generic instructions on how to create a connection in Data Transforms.

- Create an Oracle Cloud Infrastructure (OCI) Generative AI connection. See Create and use an Oracle Cloud Infrastructure Generative AI Connection.

To use vector embeddings in a data flow:

1. Follow the instructions in Create a Data Flow to create a new data flow.
2. In the Data Flow Editor click **Add a Schema** to define your source connection. From the **Connection** drop-down, select the Oracle Database 23ai connection and the schema that you want to use from the drop down. Click **OK**.
3. Drag the tables that you want to use as a source in the data flow and drop them on the design canvas.
4. From the Database Functions toolbar, click **Machine Learning** and drag the **Text Embedding Vector** transformation component drop it on the design canvas.
5. Click the **Text Embedding Vector** transformation component to view its properties.
6. In the **General** tab, specify the following:
   - **AI Service** - Select **OCI Generative AI** from the drop-down.
   - **Connection** - The drop-down lists all the available connections for the selected AI Service. Select the connection that you want to use.
   - **AI Model** - The drop-down lists all the available models for the selected AI Service and Connection. The following models are listed:
     - "cohere.embed-english-light-v2.0"
     - "cohere.embed-english-light-v3.0"
     - "cohere.embed-english-v3.0"
     - "cohere.embed-multilingual-light-v3.0"
     - "cohere.embed-multilingual-v3.0"

     You can also type in the model name.
7. In the **Column Mapping** tab, map the source column that you want to embed to the INPUT attribute of the operator. The only column available in the column mappings is `input_text`. Drag a text column from the available columns to the Expression column. This is the data that the vectors will be built on.
8. Drag the table that you want to use as a target in the data flow and drop it on the design canvas.
9. Save and execute the data flow.
   Data Transforms will build vectors for each of the rows in the source table and write that to the target table.

9.6 Component Properties

The Properties Panel displays various settings for components selected in the Design Canvas.

Depending on the component selected, you may see any of the following icons:

- General ( ) - Displays the name of the component along with its connection and schema details. You can edit some of these properties.
- Attributes ( ) - Displays the details of all the attributes associated with the component.

- Column Mapping ( ) - Allows you to map all the columns automatically. See Map Data Columns for more information.
- Preview ( ) - Displays a preview of the component. For Oracle tables, you can also view the statistics of the selected data entity. See View Statistics of Data Entities for details about the statistical information available.
- Options ( ) - Displays options such as
  - Truncate Table - Replaces any existing target table content with new data.
  - Append - Inserts records from the flow into the target. Existing records are not updated.
  - Incremental - Integrates data in the target table by comparing the records of the flow with existing records and updating the records when their associated data is not the same. Those that don't yet exist in the target are inserted.
    The option includes an Auto compression feature that is set to `True` by default. For data flow jobs that use the Incremental Update mode to load data onto a compressed Oracle target partition, the Auto compression feature recompresses the modified target partitions after the load completes successfully. For table partitions that are not originally compressed, the compression is skipped irrespective of whether Auto compression is set to true.

> **Note:**
>
> The Auto compression option is available to the ADMIN user or to a user with the DWROLE role. For data flows that have schema users other than ADMIN you need to either assign the DWROLE to the user or disable Auto compression to avoid execution errors.

9.7 Map Data Columns

When you connect the source data entity with the target data entity, the column names are automatically mapped by the column names. You have a choice to map the columns by Position or by Name or map the columns manually using the Expression Editor.

To map columns by Position or by Name:

1. Select the target Data Entity.
2. Click the arrow icon present on the top right corner to expand the Properties Panel. This will give you more space to work with.
3. In the **Properties Panel**, click the **Column Mapping** icon ( ).
4. To map the columns by Position or by Name, from the **Auto Map** drop-down menu, select **By Position** or **By Name**.

To map the columns manually:

1. From the **Auto Map** drop-down menu, select **Clear** to clear the existing mappings.
2. Drag and drop the attributes from the tree on the left to map with the Expression column.

3. To edit an expression, click the **Edit** icon of the respective column. The **Expression Editor** appears allowing you to perform the required changes (for example, you can just add an expression-"UPPER" or open the Expression Editor to edit the expression).

> **Note:**
>
> Use the expression editor only if you have complex expressions for a particular column.

4. Click **OK**.

9.8 Validate and Execute a Data Flow

After your mappings are ready, you can proceed to validate and execute the data flow.

Do the following:

1. Click Save.
   After saving, if data needs to be staged before transforming, Transfer button is added to one or more links. You can click these buttons to set more options, if available.

2. Click the **Code Simulation** icon (</>) if you want to check the code that will run to complete the tasks that are performed when you execute the data flow job. The source and target details are displayed in different colors for ease of reference. This is handy if you want to check if the mapping is correct before you run the job or if the job fails. Note that the code cannot be used for debugging. For detailed information about the job, see the Job Details page.

3. Click the **Validate** icon (✓) in the toolbar above the design canvas to validate the data flow.

4. After a successful validation, click the **Execute** icon (▶) next to the **Validate** icon to execute the data flow.
   If you have added variables to the data flow, the **Variable values** page appears that displays the list of variables that you have added to the data flow. You can choose to use the current value, the default value, or set a custom value for each variable. Note that the custom value is applied only to the current run of the data flow. The custom value does not persist for any subsequent sessions.

   A message appears that displays the execution Job ID and name. To check the status of the data flow, see the Status panel on the right below the Properties Panel. For details about the Status panel, see Monitor Status of Data Loads, Data Flows, and Workflows. This panel also shows the link to the Job ID that you can click to monitor the progress on the Jobs page. For more information, see Create and Manage Jobs.

   For data flows created using Oracle Object Storage connections, the data from the source CSV file is loaded into the target Oracle Autonomous Database. You can also export data from an Oracle Autonomous Database table to a CSV file in Oracle Object Storage.

9.9 Schedule Data Flows or Workflows

You can schedule workflows and data flows to run at specified time interval.

Each schedule allows you to specify a start date and time, and execution frequency.

To schedule a data flow or a workflow:

1. In the left pane, click **Schedules**.
2. Click **Create Schedule**.
3. From the **Resource** drop-down menu, select **Data Flow** or **Workflow** as appropriate.
4. From the **Resource Name** drop-down list, select a data flow or workflow that you want to schedule.
5. From the **Frequency** drop-down list, select the time frame in which you wish to execute the created schedule. You can schedule it On Startup, Simple, Daily, Hourly, Weekly, Monthly or Yearly. Based on the selected frequency, **Time** field appears allowing you to schedule the selected Data Flow or Workflow. For detailed information refer to the below table:

Frequency	Time Values
On Startup	NIL
Simple	Click Select Date Time icon next to **Date and Time** field, to select the required date and time in `MM/DD/YY` and `HH:mm` format.
Daily	Click the clock icon next to **Time** field, to select the required time in `HH:mm` format.
Hourly	Click up and down arrows next to **Time** field, to select the required time in `Minutes:Seconds` format.
Weekly	• **Run Every** parameter appears with check-boxes for all the days of the week. • Click the clock icon next to **Time** field, to select the required time in `HH:mm` format.
Monthly (day of the month)	• Select the required date from the **Monthly Date** drop-down box. • Click the clock icon next to **Time** field, to select the required time in `HH:mm` format.
Monthly (week day)	• From the **Monthly date** drop down box select the required value. • From the **Week day** drop-down box, select the required weekday in which you wish to schedule the job.
Yearly	• From the **Month** drop down box, select the required month in which you wish to schedule the job. • From the **Monthly date** drop down box, select the required date in which you wish to schedule the job. • Click the clock icon next to **Time** field, to select the required time in `HH:mm` format.

6. In the **Number of Attempts on Failure** text box, enter the required value or click the up or down arrows next to the text box, to select the required value. This value denotes the number of retry attempts that should happen after the schedule failure.
7. **Stop Execution After** field denotes the time after which the schedule has to stop after executing it. It can be in **Hours**, **Minutes** or **Seconds**. Select the value and its unit.
8. Select the status of the scheduled Data Flow or Workflow from the **Status** options. It can be **Active**, **Inactive** or **Active for Period**. Through **Active for Period** option you can configure the exact time frame of the created schedule. It's Starting and Ending Date with time, time interval in which you wish to run the schedule and exceptions, if any.

9. After configuring the above details, click **Save**.
   The created schedules get listed in the **Schedules** page along with the following details:
   - Resource
   - Status
   - Frequency
   - Valid or Invalid

   Click the Actions menu (⋮) of the respective schedule to perform the following operations:
   - Click **Edit** to edit the details of the created schedule.
   - Click **Disable** to disable the created schedule. The status of a schedule becomes inactive when you disable it and gets displayed in the Scheduled list. Select the schedule and click **Enable**, to enable it whenever required. You can enable it again for a specific period, through **Enable for Period** option.
   - Click **Validate** to validate the created schedule.
   - Click **Export** to export the schedule. See Export Objects.
   - Click **Delete** to delete the created schedule. Upon confirmation the created schedule is deleted.

To schedule a data flow or a workflow specific to a project:

- Click the Project title displayed on the **Projects** page.
  Project Details page appears.
- In the left pane, click **Schedules**.
  Schedules page appears. It displays all the schedules pertaining to the selected project.

  > **Note:**
  >
  > This page is project specific and displays only the Data Flows and Workflows pertaining to the selected project.

- To create a new schedule, click **Create Schedule**.
- Enter all the required details and click **Save**.
  The new schedule is created and is added to the existing list of schedules for the project.

9.10 Monitor Status of Data Loads, Data Flows, and Workflows

When you run a data load, data flow, or workflow Oracle Data Transforms runs jobs in the background to complete the request. You can view the status of the job in the panel on the bottom right of the Data Load Details, the Data Flow Editor, and the Workflow Editor page.

This panel includes the following:

- A Refresh icon (↻) to refresh the displayed status.
- The current status. You can see any of the following statuses:
  - Not Started – If no data load, data flow, or workflow has been executed yet.
  - Running – When the job starts.

- Done – When the job completes.
  The status of the last job run persists in this panel till you run a new data load, data flow, or workflow.

  > **Note:**
  >
  > For data loads created using OCI GoldenGate, the status panel shows the link to the GoldenGate Deployment Console and the status of the Extract and Replicat processes.

- The name of the job and a link that takes you to the Jobs Details page where you can see detailed information about the job and monitor the execution status. For more information, see Create and Manage Jobs.
  When you run a data load, multiple jobs run in the background to complete the request. This panel shows a link for each job that executes when running a data load.

10

Workflows

Get detailed information about creating and working with workflows.

Topics

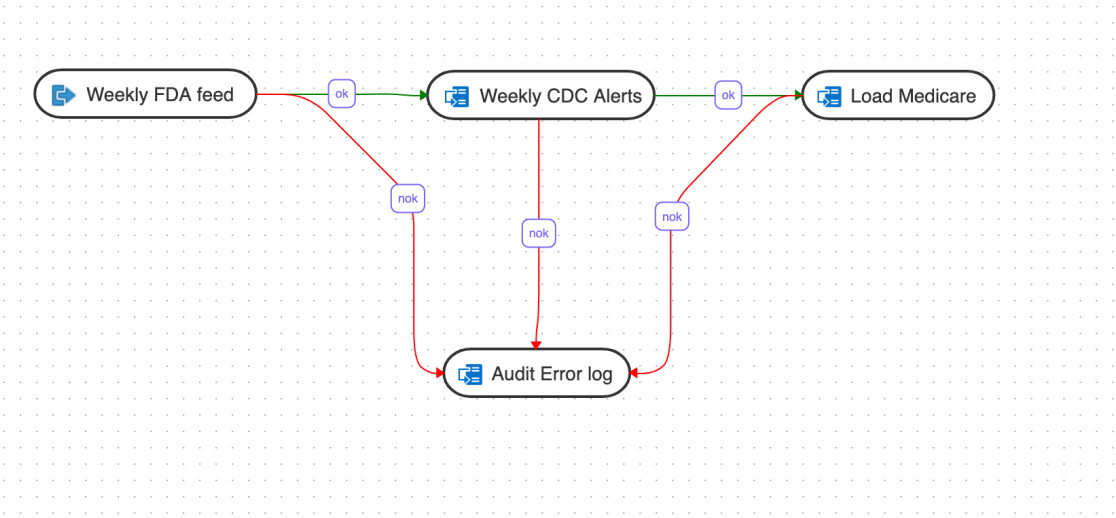
- Introduction to Workflows
  A workflow is made up of multiple flows organized in a sequence in which they must be executed.
- Create a New Workflow
  You can add data flows, workflows, variables, or data loads in a workflow.
- Define a Data Studio Data Load in a Work Flow
  You can add a Data Studio Data Load as a step in a workflow and run the data load at scheduled intervals.
- Schedule Data Flows or Workflows
  You can schedule workflows and data flows to run at specified time interval.
- Monitor Status of Data Loads, Data Flows, and Workflows
  When you run a data load, data flow, or workflow Oracle Data Transforms runs jobs in the background to complete the request. You can view the status of the job in the panel on the bottom right of the Data Load Details, the Data Flow Editor, and the Workflow Editor page.

10.1 Introduction to Workflows

A workflow is made up of multiple flows organized in a sequence in which they must be executed.

Each data flow is executed as a step. You can also add workflows, SQL queries, data loads, as well as variables as steps within a workflow. When you execute a workflow, a step either succeeds or fails. Depending on whether the first step succeeds or fails, you can choose the next step that must be executed.

Here is an example of a workflow:



In this example, the workflow performs the following actions:

1. Execute the "Weekly FDA feed" data flow.
2. If the "Weekly FDA feed" data flow execution is successful, execute the "Weekly CDC alerts" data flow.
3. If the "Weekly CDC alerts" data flow execution is successful, execute the "Load Medicare" data flow.
4. If any of the above data flow fails, then execute the "Audit_error_log" data flow.

10.2 Create a New Workflow

You can add data flows, workflows, variables, or data loads in a workflow.

To create a new workflow for your project:

1. On the **Home** page, click the required Project title.
   You are navigated to the **Project Details** page.
2. In the left pane, click **Workflows**.
   The Workflow page appears.
3. On the **Workflow** page, click **Create Workflow**.
   The **Create Workflow** page appears.
4. In the **Name** field provide a name for the new workflow and click **Create**.
   The new workflow is created and listed in the **Workflow** page with its **Name** and **Folder** details. Click the Actions icon (⋮) next to the workflow to edit, rename, copy, change folder, start, export, or delete it.
5. Click the Workflow to configure the **Workflow Details**.
6. From the left panel drag the data flows, workflows, variables, or data loads that you want to run in the workflow. To use variables in the workflow, see Use Variables in a Workflow for more information.
   If you have connected to any other Data Transforms instance, you can also add data loads that you have created in that Data Transforms instance. See Create a Data Transforms Connection for Remote Data Load and Run a Data Load for more information.
7. Select either the ok (green arrow) icon, the not ok (red arrow) or the ok/not ok (black arrow) in the toolbar.

This defines the mode that subsequent links drawn on the canvas will be set to.

- ok (green) defines the success path.
- not ok (red) defines the failure path.
- ok/not ok (black) defines a path that is followed on success or failure.

8. Use the Sleep icon (

) to add a delay in the workflow. Drag the Sleep icon on the canvas and connect it in the flow with either the ok (green), not ok (red) or ok/not ok (black) links. This will add a delay at that point in the flow.

9. If you wan to add a SQL or PL/SQL query as a step in the workflow, do the following:

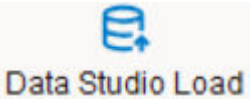
- Drag the SQL icon (

) on the canvas.
- Double click the SQL step in the editor to open the step properties page.
- Select the **Attributes** tab.
- From the **Connection** drop-down select the connection you want to run the query on.

> **Note:**
>
> The drop-down lists only Oracle database connections.

- In the **SQL** textbox add the query that you want to run.

10. If you want to add a Data Studio Data Load as a step in the workflow, use the Data Studio Load icon (



) on the canvas. See Define a Data Studio Data Load in a Work Flow for detailed instructions on how to use a Data Studio Data Load to load data to Autonomous Database.

11. Select the step and click the **Connector** icon () next to it to connect it with the next step.

12. After defining all the required Workflow details,

- Click  , to save the created/designed workflow.
- Click  , to align the nodes of the designed workflow.
- Click  , to execute the created workflow.
  If you have added variables to the workflow, the **Variable values** page appears that displays the list of variables that you have added to the workflow. You can choose to

use the current value, the default value, or set a custom value for each variable. Note that the custom value is applied only to the current run of the workflow. The custom value does not persist for any subsequent sessions.

- Select a single step in the canvas and click the **Execute Step** icon (), to execute only the selected data flow or workflow.
  To check the status of the workflow, see the Status panel on the right below the Properties Panel. For details about the Status panel, see Monitor Status of Data Loads, Data Flows, and Workflows. This panel shows the link to the Job ID that you can click to monitor the execution status on the Jobs page. For more information about jobs see Create and Manage Jobs.

- Click   , to maximize or minimize the created workflow diagram in the design canvas.
  The newly created workflows get listed in the Project Details page. The following details are displayed:
  - Name of the Workflow
  - Folder corresponding to the workflow - Default Folder

  To view the details of a workflow, click the name of the workflow and you are navigated to the Workflow Details page.

In the left pane, you can search for the required Data Flow or Workflow using the Name filter. In the **Name** text box, enter the name of the required Data Flow or Workflow.

Select a step in the canvas and check the **Properties Panel** available at the right side of the design canvas to know the following properties of the selected step in a created Data Flow or Workflow:

- Name
- Linked Object

> **Note:**
>
> You cannot edit this field.

- Step -
  - First Step - Select the **First Step** check-box, to execute the selected step as the first step for execution in a Data Flow or Workflow.

> **Note:**
>
> You can select only a single step as a first step for execution in a Data Flow or Workflow.

- Number of attempts on Failure
- Time between attempts in seconds(s)
- Log steps in journal - Select any of the following options from the drop-down box:
  - Always
  - Never

– Error

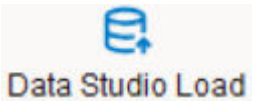
10.3 Define a Data Studio Data Load in a Work Flow

You can add a Data Studio Data Load as a step in a workflow and run the data load at scheduled intervals.

To do this, you need to create a connection, import data entities, create a data flow and then schedule the data load run.

To define and run a Data Studio Data Load within a workflow:

1. On the **Home** page, click the required Project title.
   You are navigated to the **Project Details** page.
2. In the left pane, click **Workflows**.
   The Workflow page appears.
3. On the **Workflow** page, click **Create Workflow**.
   The **Create Workflow** page appears.
4. In the **Name** field provide a name for the new workflow and click **Create**.
   The new workflow is created and listed in the **Workflow** page with its **Name** and **Folder** details. Click the Actions icon (⋮) next to the workflow to edit, rename, copy, change folder, start, export, or delete it.
5. Click the Workflow to configure the **Workflow Details**.
6. To add a Data Studio Data Load as a step in the workflow, drag the Data Studio Load icon (

) on the canvas.
7. Double click the Data Studio Load step in the editor to open the step properties page.
8. Select the **Attributes** tab.
9. From the **Connection** drop-down select the connection you want to load the data from.

   > **Note:**
   >
   > The drop-down lists only Oracle database connections.

10. Select a Data Studio Load from the drop down.
11. Click , to save the created/designed workflow.
12. Click , to execute the created workflow.
13. To check the status of the workflow, see the Status panel on the right below the Properties Panel. For details about the Status panel, see Monitor Status of Data Loads, Data Flows, and Workflows. This panel shows the link to the Job ID that you can click to monitor the execution status on the Jobs page. For more information about jobs see Create and Manage Jobs.

The newly created workflows get listed in the Project Details page.

10.4 Schedule Data Flows or Workflows

You can schedule workflows and data flows to run at specified time interval.

Each schedule allows you to specify a start date and time, and execution frequency.

To schedule a data flow or a workflow:

1. In the left pane, click **Schedules**.
2. Click **Create Schedule**.
3. From the **Resource** drop-down menu, select **Data Flow** or **Workflow** as appropriate.
4. From the **Resource Name** drop-down list, select a data flow or workflow that you want to schedule.
5. From the **Frequency** drop-down list, select the time frame in which you wish to execute the created schedule. You can schedule it On Startup, Simple, Daily, Hourly, Weekly, Monthly or Yearly. Based on the selected frequency, **Time** field appears allowing you to schedule the selected Data Flow or Workflow. For detailed information refer to the below table:

Frequency	Time Values
On Startup	NIL
Simple	Click Select Date Time icon next to **Date and Time** field, to select the required date and time in `MM/DD/YY` and `HH:mm` format.
Daily	Click the clock icon next to **Time** field, to select the required time in `HH:mm` format.
Hourly	Click up and down arrows next to **Time** field, to select the required time in `Minutes:Seconds` format.
Weekly	• **Run Every** parameter appears with check-boxes for all the days of the week. • Click the clock icon next to **Time** field, to select the required time in `HH:mm` format.
Monthly (day of the month)	• Select the required date from the **Monthly Date** drop-down box. • Click the clock icon next to **Time** field, to select the required time in `HH:mm` format.
Monthly (week day)	• From the **Monthly date** drop down box select the required value. • From the **Week day** drop-down box, select the required weekday in which you wish to schedule the job.
Yearly	• From the **Month** drop down box, select the required month in which you wish to schedule the job. • From the **Monthly date** drop down box, select the required date in which you wish to schedule the job. • Click the clock icon next to **Time** field, to select the required time in `HH:mm` format.

6. In the **Number of Attempts on Failure** text box, enter the required value or click the up or down arrows next to the text box, to select the required value. This value denotes the number of retry attempts that should happen after the schedule failure.
7. **Stop Execution After** field denotes the time after which the schedule has to stop after executing it. It can be in **Hours**, **Minutes** or **Seconds**. Select the value and its unit.
8. Select the status of the scheduled Data Flow or Workflow from the **Status** options. It can be **Active**, **Inactive** or **Active for Period**. Through **Active for Period** option you can configure the exact time frame of the created schedule. It's Starting and Ending Date with time, time interval in which you wish to run the schedule and exceptions, if any.
9. After configuring the above details, click **Save**.
   The created schedules get listed in the **Schedules** page along with the following details:
   - Resource
   - Status
   - Frequency
   - Valid or Invalid

   Click the Actions menu (⋮) of the respective schedule to perform the following operations:
   - Click **Edit** to edit the details of the created schedule.
   - Click **Disable** to disable the created schedule. The status of a schedule becomes inactive when you disable it and gets displayed in the Scheduled list. Select the schedule and click **Enable**, to enable it whenever required. You can enable it again for a specific period, through **Enable for Period** option.
   - Click **Validate** to validate the created schedule.
   - Click **Export** to export the schedule. See Export Objects.
   - Click **Delete** to delete the created schedule. Upon confirmation the created schedule is deleted.

To schedule a data flow or a workflow specific to a project:

- Click the Project title displayed on the **Projects** page.
  Project Details page appears.
- In the left pane, click **Schedules**.
  Schedules page appears. It displays all the schedules pertaining to the selected project.

  > **Note:**
  >
  > This page is project specific and displays only the Data Flows and Workflows pertaining to the selected project.

- To create a new schedule, click **Create Schedule**.
- Enter all the required details and click **Save**.
  The new schedule is created and is added to the existing list of schedules for the project.

10.5 Monitor Status of Data Loads, Data Flows, and Workflows

When you run a data load, data flow, or workflow Oracle Data Transforms runs jobs in the background to complete the request. You can view the status of the job in the panel on the bottom right of the Data Load Details, the Data Flow Editor, and the Workflow Editor page.

This panel includes the following:

- A Refresh icon ( ) to refresh the displayed status.
- The current status. You can see any of the following statuses:
  - Not Started – If no data load, data flow, or workflow has been executed yet.
  - Running – When the job starts.
  - Done – When the job completes.
    The status of the last job run persists in this panel till you run a new data load, data flow, or workflow.

    > **Note:**
    >
    > For data loads created using OCI GoldenGate, the status panel shows the link to the GoldenGate Deployment Console and the status of the Extract and Replicat processes.

- The name of the job and a link that takes you to the Jobs Details page where you can see detailed information about the job and monitor the execution status. For more information, see Create and Manage Jobs.
  When you run a data load, multiple jobs run in the background to complete the request. This panel shows a link for each job that executes when running a data load.

11 Variables

A variable is an object that stores a single value, which can be a string, a number or a date.

Topics

- Create a Variable
  You can specify the value of the variables when you create the variable or set the value using a query expression.
- Use Variables in a Data Flow
  You can use variables in a data flow by using them as a filter definition. You can refer to variables in a data flow using the format `#<variable_name>`. During execution the variable is substituted by the value.
- Use Variables in a Workflow
  The left panel of the Workflow Details page lists the variables that you can use as steps within a workflow.

11.1 Create a Variable

You can specify the value of the variables when you create the variable or set the value using a query expression.

The Variables page lists all the existing variables. Click the **Actions** icon (⋮) of a variable to edit, refresh, or delete it. All the available variables are also listed in the left panel of the Workflow Details page. You can use variables as steps in data flows and workflows.

To create a variable:

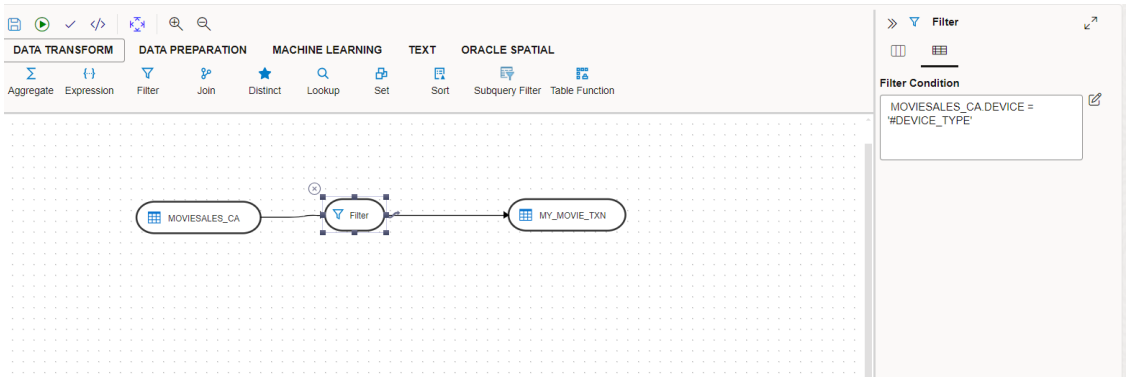
1. On the **Home** page, click the required Project title. You are navigated to the **Project Details** page.
2. In the left pane, click **Variables**. The Create Variable page appears.
3. Enter a name for the variable.
4. From the **Data Type** drop down, select one of the following:
   - **Short text** - The value can be alphanumeric, can contain special characters, and cannot exceed 255 characters.
   - **Long text** - The value can be alphanumeric, can contain special characters, and cannot exceed 64000 characters.
   - **Numeric** - The value can be a numeric value and can be preceded by a minus (-) sign. The value cannot exceed 10 digits. Possible value range is -999999999 to 9999999999.
   - **Date** - The value is a date format.
5. In the **Default Value** field enter the value you want to assign to the variable. The allowed value depends on the **Data Type** you select.

6. For variables that you modify, you can click the Actions menu next to the variable to open the Refresh Variable page to update the value. From the **Keep History** drop down select how you want the variable value to be displayed in the Refresh Variable page:
   - **All Values** - You can see the history of all the values held by this variable.
   - **Latest Value** - You can see only the latest value specified for the variable.
   - **No History** - Data Transforms does not keep the history of the values held by this variable.
   - **Secure Value** - This is useful when the variable contains passwords or other sensitive data. The value is not displayed in the Refresh Variable page.
7. Enter a description for the variable.
8. If you want the variable value to be set by a query, click the **Refresh** tab. Select the connection type and schema where you want to execute the command. Enter the query and click **Validate** to check the syntax of your expression.
9. Click **Save**.
   The newly created variable is listed in the Variables page as well as in the Variables node in the left panel of the Workflow Details page.

11.2 Use Variables in a Data Flow

You can use variables in a data flow by using them as a filter definition. You can refer to variables in a data flow using the format `#<variable_name>`. During execution the variable is substituted by the value.

When you execute the data flow, the **Variable values** page appears that displays the list of variables that you have added to the data flow. You can choose to use the current value, the default value, or set a custom value for each variable. Note that the custom value is applied only to the current run of the data flow. The custom value does not persist for any subsequent sessions.
Here is an example of the use of a variable in a data flow:



In this example, the data flow uses the variable DEVICE_TYPE as a step with the filter condition set as follows:

```
MOVIESALES_CA.DEVICE = '#DEVICE_TYPE'
```

The variable is quoted because the substituted value needs to be quoted. Since the variable value is substituted during execution, you can use it for many use cases such as in the transformation expression in the mapping.

11.3 Use Variables in a Workflow

The left panel of the Workflow Details page lists the variables that you can use as steps within a workflow.

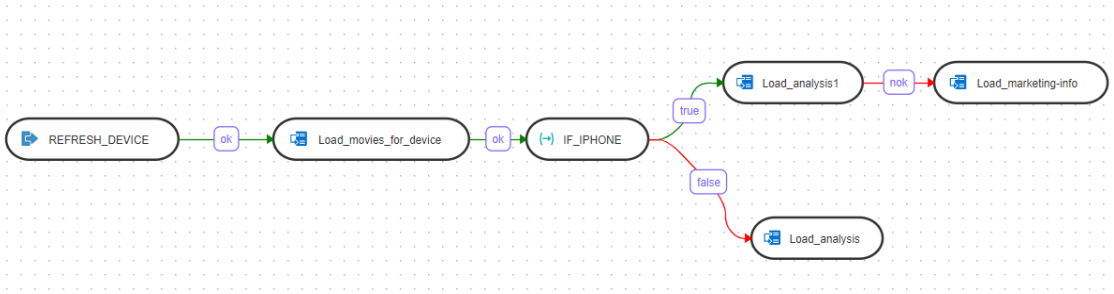
You can drag and drop the variables you want to use in the workflow on the design canvas. The Properties Panel available on the right side of the design canvas displays various details about the step such as the following.

- **Name**, **Type**, and the **Step** sequence number. These are non-editable values.
- Click the link under **Linked Object** to view and update the details about the variable.
- **Operation**: Select any one from the following:
  - **Set Variable** - There are two functions for this step:
    Update sets the current value of a variable.

    Increment increases or decreases a numeric value by the specified amount.
  - **Refresh Variable** - This variable step refreshes the variable by running the query specified in the variable definition.
  - **Evaluate Variable**: This variable step type compares the value of the variable with a given value according to an operator. If the condition is met, then the evaluation step is true, otherwise it is false.
- **Number of attempts on Failure**
- **Time between attempts in seconds(s)**
- **Log steps in journal** - You can select from Always, Never, or Error.

When you execute the workflow, the **Variable values** page appears that displays the list of variables that you have added to the workflow. You can choose to use the current value, the default value, or set a custom value for each variable. Note that the custom value is applied only to the current run of the workflow. The custom value does not persist for any subsequent sessions.

Here is an example of the use of a variable in a workflow:



In this example, a variable called DEVICE _TYPE with the value "iphone" is used as a step in the workflow. The workflow performs the following actions:

1. Execute the "REFRESH_DEVICE data flow.
2. If the "REFRESH_DEVICE" data flow execution is successful, execute the "Load_movies_for_device" data flow.

3. If the "Load_movies_for_device" data flow execution is successful, execute the "IF_IPHONE" variable evaluation step.
4. If the variable value is equal to ‘iphone’, execute path for “true”.
5. If the variable value is not equal to ‘iphone’, execute path for “false”.

12

Machine Learning (ML) Models

Data Transforms supports the use of ML Model in a data flow. Learn how to create and use Machine Learning (ML) Models in data flows.

Topics

12.1 Create an ML Model Data Entity in the Data Flow editor

To use ML models in Data Transforms you need to create two data flows. You need to first build the ML model data entity using the Data Flow editor, and then you can use the data entity in a data flow to mine data from a source connection and load it into a target server.

To build an ML Model data entity in the Data Flow editor,

1. Drag the data entity that you want to build the ML Model on onto the Design Canvas.
2. Select the component and click the Add Data Entity icon present on the top right corner of the target component.
3. **Add Data Entity** page appears allowing you to configure the following details of the target component:
   General tab
   - In the **Name** text box, enter the name of the newly created Data Entity.
   - From the **Entity Type** drop-down, select **ML Model** as the data entity type.
     When you select this entity type the user interface changes as follows:
     – The **Connection** drop down only lists the Oracle connections that you have created.
     – The Add Data Entity wizard displays the **Properties** tab where you can select the Type of Learning, Function, Algorithm, and configure parameters to define the ML Model. See ML Model Data Entity Properties for more information.
   - From the **Connection Type** drop-down, select the required connection from which you wish to add the newly created Data Entity. For ML Model data entities, the **Connection Type** drop-down only lists **Oracle** as the option.

- The Connection drop-down is populated with the connections you have created with the associated connection type. From the **Connection** drop-down, select the server name where you wish to keep the ML model data entity.
- In the **Schema** drop-down, all schema corresponding to the selected connection are listed in two groups.
  - New Database Schema (ones that you've not imported from before) and
  - Existing Database Schema (ones that you've imported from before and are potentially replacing data entities).

  From the **Schema** drop-down, select the required schema.
- In the **Tags** text box, enter a tag of your choice. You can use tags to filter the Data Entities displayed in the Data Entity Page.
- If you want to mark this data entity as a feature group, expand **Advanced Options** and click the **Treat as Feature Group** checkbox.
- Click **Next**.

Properties tab

- Select the **Type of Learning**, **Function**, and **Algorithm** you want to use to build this data entity. For more information about the options, see ML Model Data Entity Properties.
- Based on the options selected, the **Parameters** section is populated with the list of parameters that are marked as "Importance" and "High". You can add other required parameters using the  icon.
  You must specify a value for each parameter so that the data flow can run successfully.

Columns tab

- Click the  Add Columns icon, to add new columns to the newly created Data Entity.
  A new column is added to the displayed table.
- The table displays the following columns:
  - Name
  - Data Type - Click the cell to configure the required Data Type.
  - Scale
  - Length
  - Actions - Click the cross icon to delete the created column.
- To delete the columns in bulk, select the columns and click the  Delete icon.
- To search for the required column details, in the Search text box enter the required column name and click enter. The details of the required column are displayed.
- Click **Next**.

Preview Data Entity tab

It displays a preview of all the created columns and their configured details. If the data entity belongs to an Oracle database, you can also view statistics of the table. See View Statistics of Data Entities for more information.

4. Click **Save** to save the configuration and exit the wizard.
5. Save and execute the data flow.
   The new Data Entity is created. displayed in the Data Entities page.

12.2 ML Model Data Entity Properties

The **Properties** tab of the **Add Data Entity** wizard provides data mining options that you can use to define the ML Model data entity.

This topic assumes prior knowledge of Oracle Machine Learning concepts such as data mining functions and algorithms. For more information, see Oracle Machine Learning for SQL API Guide.

Type of Learning	Function	Algorithm
Supervised	Classification	Decision Tree
		Explicit Semantic Analysis
		Generalized Linear Models
		Naive Bayes
		Random Forest
		Neural Network
		Support Vector Machines
	Regression	Generalized Linear Models
		Neural Network
		Support Vector Machines
	Time Series	Exponential Smoothing
	Attribute Importance	Minimum Description Length
Unsupervised	Association	Apriori
	Attribute Importance	CUR matrix decomposition
	Anomaly Detection	One Class Support Vector Machines
	Clustering	Expectation Maximization
		k-Means
		Orthogonal Partitioning Clustering
	Feature Extraction	Explicit Semantic Analysis
		Non-Negative Matrix Factorization
		Singular Value Decomposition

12.3 Use ML Model in a Data Flow

You can use the **Prediction Model** database function to run ML Model algorithms on source data and load the output to a target database.

Before you use an ML Model in a data flow, you need build the ML Model. For instructions on how to create an ML model, see Create an ML Model Data Entity in the Data Flow editor.

To use an ML Model in a data flow:

1. Follow the instructions in Create a Data Flow to create a new data flow.
2. In the Data Flow Editor, drag the tables that you want to use as a source in the data flow and drop them on the design canvas.
3. From the Database Functions toolbar, click **Machine Learning** and drag the **Prediction Model** transformation component drop it on the design canvas.
4. Click the **Prediction Model** transformation component to view its properties.
5. In the **General** tab, specify the following:
   - **Connection** - The drop-down lists all the available Oracle connections. Select the Oracle connection that you want to use.
   - **Schema** - Select the schema.
   - **ML Model** - The drop-down lists all the available ML models. See Create an ML Model Data Entity in the Data Flow editor for instructions on how to build an ML Model.
6. In the **Column Mapping** tab, map the source column that you want to embed to the INPUT attribute of the operator. The only column available in the column mappings is `prediction parameters`. Drag a text column from the available columns to the Expression column.
7. Drag the table that you want to use as a target in the data flow and drop it on the design canvas.
8. Save and execute the data flow.
   Data Transforms will run the prediction model on the source data and write the output to the target table.

13
Jobs

This section includes information about creating and working with jobs.

Topics

- Create and Manage Jobs
  When you execute a data load, data flow or workflow Oracle Data Transforms creates a job to complete the process in the background.

13.1 Create and Manage Jobs

When you execute a data load, data flow or workflow Oracle Data Transforms creates a job to complete the process in the background.

A job is made up of multiple steps that corresponds to an execution task.

The **Jobs** page by default, lists the execution job sessions that are running and completed for the present day as `Date` parameter is set to `Today` by default. You can click on any of the instance IDs from the list to see its details.

To work with jobs in Data Transforms, you can do the following:

- Create a Job from the Home page
- Create a Job from the Projects page
- View and Delete a Job
- Configure Purge Interval for Jobs
- Search for a Job

Create a Job from the Home page

To create a new job,

- On the Home page, click **Jobs** on the left pane to access the Jobs page.
- Click **Create Job**.
  The **Create Job** page slides-in.
- From the **Resource Type** drop-down, select the type of resource for which you wish to create a new job. It can be - Data Flow, Workflow or Schema.
- All the resources associated with the selected Resource Type get listed in the **Resource** field. From the **Resource** drop-down, select the required resource for which you wish to create a new job.
- Click **Create**.

A new job is created and is added to the existing list of jobs in the Jobs page.

For each session, you can see the Job Session **ID**, **Name**, **Status**, **Start Time** and **End Time** for each of the jobs that are executed.

Create a Job from the Projects page

To view the job sessions pertaining to a specific project:

- Click the Project tile displayed in the Projects page.
  Project Details page appears.
- In the left pane, click **Jobs**.
  Jobs page appears. It displays all the jobs and their details pertaining to the selected project.

> **Note:**
>
> This page is project specific and displays only the jobs pertaining to the selected project.

- To create a new job, click **Create Job**.
- Enter all the required details and click **Create**.
  A new job is created and is added to the existing list of jobs for the project.

View and Delete a Job

Click the Actions menu (⋮) to **View Details** of the jobs, **Delete** and **Rerun** the jobs, if required. If any of the steps failed, you can click on them to see the error details.

To delete a job, select the check box of the respective job session and click **Delete**. Upon confirmation the selected job gets deleted.

Configure Purge Interval for Jobs

You can configure the interval to purge older jobs to reduce the load on the sessions table. The default time interval is 30 days, which means that all jobs that are older than 30 days will be purged periodically. Note that only an ADMIN user (odiadmin) can configure the purge interval.

To configure the purge interval for jobs, click the gear icon () on the Jobs page. In the **Configurations** dialog box, set the required value in the **Delete jobs before (days)** field and click **Save**.

Search for a Job

You can also search for the required job session to know its details based on filters such as:

- **Name** - Name of the Job
- **Status** - Select the required session status from the list - All, Done, Error, Queued, Running, Waiting and Warning. Note that to use the Queued and Waiting filter options in the drop-down you must enable the **Show Queued/Waiting** Jobs option.
- **Show Queued/Waiting Jobs** – Enable this option to include the queued or waiting jobs in the list on the Jobs page. When this option is enabled, the **Status** drop-down includes the Queued and Waiting options that you can use to filter the list.
- **Date** - Select the date in which the required job session was executed - All, Today, Yesterday, Last Week, Last Month and Custom Range, which allows you to select specific From and To dates.

14
Users

APPLIES TO: Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

Learn about creating and managing users.

Topics

14.1 Create Users

APPLIES TO: Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

The User Management page of the Administration tab of Oracle Data Transforms helps you to manage users.

A user corresponds to the login name used to connect to a repository, and can have privileges depending on the role assigned. The SUPERVISOR user is assigned the Oracle Data Integrator Admin (odiadmin) role by default. The odiadmin role allows users to:

- Assign the odiadmin role to a user
- Change user password
- Delete users

The SUPERVISOR user can assign this role when creating new users.

To create a new user:

1. In the left pane, click **Administration**.
   A warning message appears.
2. Click **Continue**.
3. In the left pane, click **User Management**.
   User Management screen appears, which displays the list of users.
4. Click **Create User**.
5. In the Create User page, enter the name. Maximum length is 50 characters.
6. Select the **Allow managing other users** checkbox to allow this user to reset passwords or delete users.

> **Note:**
>
> This option is only available if you are the SUPERVISOR user or have been assigned the odiadmin role.

7. Enter and reenter and password for the new user.

> **Note:**
>
> The password should:
>
> - contain alpha-numeric characters,
> - be 6-12 characters in length,
> - not start with a number,
> - contain at least one alphabet in upper case,
> - contain at least one special character. Only $, # and _ are allowed.

8. Click **Create User**.
   The newly created user appears in the list.

14.2 Change User Passwords

APPLIES TO: Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

You must have the odiadmin role assigned to be able to change the password of any of the listed users. If you are not assigned the odiadmin role, you can only change the password of the logged in user.

In this case, the **Actions** menu is available only for the logged in user. You cannot change the password of any of the other users, even the ones that you have created.

To change the user password:

1. On the User Management page, click **Change Password** from the user's Actions icon (⋮).
2. Enter the new password and click **Update User**.

To change the password of the SUPERVISOR user:

1. On the User Management page, click **Change Password** from the user's Actions icon (⋮).
2. Enter the new password and click **Update User**.
   You are logged out of Data Transforms.
3. Restart the jetty server. Log in as OPC user and execute the following commands:

```
ssh -i <path to id_rsa>  opc@<Instance IP>
sudo su
```

```
systemctl stop|start jettyserver.service
exit
```

4. Launch Oracle Data Transforms from the Database Actions page.

14.3 Delete Users

APPLIES TO: Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

You must have the odiadmin role assigned to be able to delete any of the listed users (except the SUPERVISOR).

To delete a user, you need to assign ownership of the objects created by this user to any of the other listed users.

> **Note:**
>
> You cannot delete users that have jobs in the running state.

If you are not assigned the odiadmin role, the **Delete** option is not available in the Actions menu.

To delete a user:

1. On the User Management page, click **Delete** from the user's Actions icon (⋮).
2. Select a new owner for the objects created by this user from the **Other Users** drop-down.
3. Click **Delete**.

15

Export and Import Objects

You can move Data Transforms objects between environments. You require an Object Storage connection to store the exported file. You then need to use the same Object Storage connection to import the objects.

Topics

15.1 Export Objects

You can export Data Transforms artifacts such as projects, connections, data loads, data flows, workflows, and schedules from one environment and import them to another.

The Export and Import options in Data Transforms allow you to do the following:

- Move code from development environments to production environments.
- Share code between development environments.
- Backup and restore objects.

To export objects from a Development environment you need to provide an Object Storage connection as an input. You need to use the same Object Storage connection when you want to run the import process in another environment. See Import Objects. The export operation exports all the metadata of the repository objects along with all its dependent parent and child objects, and stores the information in an XML format in the Object Storage bucket.

You can use the Export wizard to export the entire project or export individual objects at the following levels:

- Projects
- Connections
- Data Load
- Data Flow
- Workflow
- Schedule

When you export an individual object, all the dependent objects are automatically included in the export.

> **Note:**
>
> Individual variables export is not currently supported. However, when you export a Project, the variables within that project are also exported.

To export objects using the Export wizard:

1. From the left pane, click the Home tab. Click **Export**.
   The **Export** wizard appears.
2. In the **Export File Name** field, enter a name for the file. The exported data will be stored in the ZIP format with the *filename_timestamp_DTR* file naming convention.
3. Select an Object Storage connection from the drop-down to store the exported file. Note that you need to select the same Object Storage connection during the import operation.
4. Click **Next**.
5. Select the objects that you want to export. You can choose to export a single, multiple, or all the objects that are listed on the page.
6. Click **Next**.
7. Select the connections that you want to export.
   You may choose not to make any selection on this page. The connections that are associated with the objects that you selected in the previous screen are automatically included during the export operation.
8. Click **Next** to preview your selections.
9. Click **Export** to start the export operation.
   A confirmation prompt appears when the export operation starts. Click the link that is displayed in the prompt to monitor the progress of the export operation on the Job Details page.

To export individual objects:

1. Click the **Actions** menu of the project, connection, dataload, dataflow, workflow, or schedule that you want to export, and click **Export**.
2. In the **Export File Name** field, enter a name for the file. The exported data will be stored in the ZIP format with the *filename_timestamp_DTR* file naming convention.
3. Select an Object Storage connection from the drop-down to store the exported file. You need to select the same Object Storage connection during the import operation.
4. Click **Export** to start the export operation.
   A confirmation prompt appears when the export operation starts. Click the link that is displayed in the prompt to monitor the progress of the export operation on the Job Details page.

15.2 Import Objects

To import objects you need to provide the same Object Storage connection that you had used for the export operation.

Note the following about the import process:

- **Projects** - To import projects, you can choose whether you want to overwrite the data in the repository or merge the matching container objects with the objects that are being imported.

  - **Overwrite** - If there is a match in the repository, all the dependent child objects will be overwritten by default.
    If there is no match, the project along with the child objects will be copied into the repository.
  - **Merge** - If the project exists in the repository, but the dependent child objects have no match, all the child objects in the repository will be merged with the new child objects that are coming through the import process.
    If the project exists in the repository, and the dependent child objects have a match, all the child objects in the repository will be overwritten with the new child objects that are coming through the import process.

    If the project does not exist in the repository, the project along with the child objects will be copied into the repository.

- **Connections** - For connections that have a match in the repository, the import process updates only the advanced properties of the connection in the repository along with the connection name. Credentials such as the schema name, passwords, and connection URL are persisted as is in the new environment. You can use such connections without any modification in the target repository.
  For connections that have no match in the repository, credentials such as the schema name, passwords, and connection URL are not carried over during the import operation. Post import, you need to update the credential information for such connections. You also need to make sure that you create a schema user in the database along with the required tables created for the schema user, which you may use as a data entity in dataloads, dataflows, workflows, and schedules.
- **Data Loads**, **Data Flows**, and **Workflows** - For these objects if there is a match in the repository, all the objects will be overwritten by default.
- **Schedule** - The status for schedules that have no match in the repository are set to Inactive. You need to update the connection information that is associated with the schedule, and then set the status of the schedule to Active.

To import objects:

1. From the left pane, click the Home tab. Click **Import**.
   The **Import** dialog box appears.
2. From the **Object Storage Connection** drop-down, select the connection that you had used for the export operation. See Export Objects.
3. From the **Import File Name** drop-down, select the exported file.
4. Choose one of the following import options:
   - **Merge** - Choose this to merge the objects that are already in the repository with the objects that are coming through the import process. This option is only available for container objects such as Projects.
   - **Overwrite** - Choose this to replace the objects that are already in the repository with the objects that are coming through the import process.
5. Click **Import** to start the import operation.
   A confirmation prompt appears when the import operation starts. Click the link that is displayed in the prompt to monitor the progress of the import operation on the Job Details page.

16

Autonomous Database

Learn about how to access Oracle Data Transforms from Oracle Autonomous Database.

Topics

16.1 Manually Register the Created Oracle Data Transforms Instance to Autonomous Database

APPLIES TO: ● Data Transforms instance that is registered with Autonomous Database.

The `RegisterODI.sh` script allows you register your Oracle Data Transforms instance with Autonomous Database.

To manually register the Oracle Data Transforms instance to Autonomous Database after creating your instance,

- Connect to the compute instance running Oracle Data Transforms as described in Connect to Oracle Data Transforms Instance.
- Navigate to your Data Transforms desktop and double click the **Register with ADP** shortcut icon.

  > **Note:**
  >
  > If you are using oel8, click **Activities** on the top left corner and click **Show Applications** to see the **Register with ADP** shortcut icon.

  or

  Execute the `RegisterODI.sh` file using the following commands:

  ```
  cd u01/oracle/transforms_home/common/scripts
  sh RegisterODI.sh
  ```

- A terminal window appears allowing you to enter the following details of the Autonomous Database instance which you had already created in Create the Oracle Data Transforms Instance step:

- Admin username - User name of the Autonomous Database instance.
- Admin password - Password of the Autonomous Database instance.
- URL Scheme - Press Enter to select the default value - http.
- IP Address - IP address of the computer which you wish to register to Autonomous Database.

> **Note:**
>
> You are prompted to provide all the above details repeatedly until you provide a valid IP address.

- Oracle Data Transforms Port - It is the port in which Oracle Data Transforms runs on the local computer. Press Enter to select the default value - 9999.

- Map the database user (created during ODI provisioning) to Oracle Data Transforms User. See Map Autonomous Database User to Data Transforms User for details.

After configuring the above details, the created instance is successfully registered to the Autonomous Database. After successful registration, you can launch Oracle Data Transforms directly from the Autonomous Database, Database Actions page. For more details on this, refer to Access Oracle Data Transforms From Database Actions page.

In some cases, you may encounter an error ("`401 Not Authorised`") when launching Oracle Data Transforms from the Database Actions page. To resolve this issue, you will need to restart the Jetty server.

To stop, start, or restart the jetty server, login as OPC user and execute the following commands:

```
ssh -i <path to id_rsa>  opc@<Instance IP>
sudo su
systemctl stop|start jettyserver.service
exit
```

16.2 Map Autonomous Database User to Data Transforms User

APPLIES TO: Data Transforms instance that is registered with Autonomous Database.

The Administration tab of Oracle Data Transforms helps you to map the database user (created during provisioning) to Oracle Data Transforms User.

The administrator in Oracle Data Transforms is called the SUPERVISOR. By mapping the users, you can directly access the Oracle Data Transforms application from Database Actions page. For more details, refer to Access Oracle Data Transforms From Database Actions page.

For mapping the users,

1. In the left pane, click **Administration**.
   A warning message appears.
2. Click **Continue**.
   User Mapping screen appears.
3. In the **Database URL** text box, enter the Database URL created during Autonomous Database instance creation, for example - `https://u6dx8gx07phfkil-db202003100909.adb.us-phoenix-1.oraclecloudapps.com/`.

To get this URL,

- From OCI console, click the Hamburger icon and navigate to **Autonomous Database** and select the autonomous database instance for which you created the repository.
- Click the **Service Console** and navigate to **Development → SQL Developer**. You can find the database URL here and copy it to the **Database URL** text box.

4. Click the Add icon () to add a new column in the User Mapping table.
5. Double click the **Database User** column to create a user mapping for the required Database User.
6. Double click the **ODI User** column and click the drop-down arrow to list all the ODI users available for mapping in Oracle Data Transforms application.
7. From the **ODI User** drop-down select the required ODI user, to whom you wish to map the newly added Database user.
8. Click **Save**.
   A confirmation message regarding the user mapping update appears.
9. Click the Refresh icon () next to the + icon, to see the newly created user mapping column in the User Mapping table.
10. If you wish to delete the created user mapping, select the user and click the Action icon ( ) next to the user and select **Delete**.
11. For deleting user mappings in bulk, select the respective user check-boxes and click **Delete**.
    Upon confirmation the selected user mappings are deleted.

You can search for the required user mapping from the User Mapping table, through Name filter.

- Enter the name of the user mapping that you are looking for in the **Name** text box.
  The respective user mapping column is displayed.

16.3 Unregister the Oracle Data Transforms Instance from Autonomous Database

APPLIES TO: Data Transforms instance that is registered with Autonomous Database.

The `UnregisterODI.sh` script allows you unregister your Oracle Data Transforms instance from Autonomous Database.

To unregister your Oracle Data Transforms instance from Autonomous Database,

- Connect to the compute instance running Oracle Data Transforms as described in Connect to Oracle Data Transforms Instance.
- Navigate to your Data Transforms desktop and double click the **Unregister with ADP** shortcut icon.

> **Note:**
>
> If you are using oel8, click **Activities** on the top left corner and click **Show Applications** to see the **Unregister with ADP** shortcut icon.

or

Execute the `UnregisterODI.sh` file using the following commands:

```
cd u01/oracle/transforms_home/common/scripts
sh UnregisterODI.sh
```

- A terminal window appears allowing you to enter the following details of the Autonomous Database instance which you had already created in Create the Oracle Data Transforms Instance step:
  - Admin username - User name of the Autonomous Database instance.
  - Admin password - Password of the Autonomous Database instance.

After providing the above details, the instance is successfully unregistered from the Autonomous Database. The Oracle Data Transforms will no longer be available in the Autonomous Database, Database Actions page.

16.4 Access Oracle Data Transforms From Database Actions page

APPLIES TO: Data Transforms instance that is registered with Autonomous Database.

The Autonomous Database users can access the Oracle Data Transforms interface from the Data Tools tab present in Database Actions page, if the DB hosts the Oracle Data Transforms repository.

Follow the below steps to access Oracle Data Transforms:

1. On the OCI console, click the Hamburger icon present on the top left corner.
2. Click **Autonomous Data Warehouse**.
   You are navigated to Autonomous Databases in ODI Compartment page.
3. Select the Autonomous Database used for creating the ODI repository (for example, `ODIQAADW1`).
   You are navigated to the respective Autonomous Database Details page.
4. From the Autonomous Database Details page, click the **Service Console** tab.
   You are navigated to the Autonomous Data Warehouse page.
5. From the left pane of the Autonomous Data Warehouse page, click **Development**.
6. In this page, select **SQL Developer Web**.
   The Oracle Database Actions login page appears.
7. In the **Oracle Database Actions** login page, in the respective **Username** and **Password** fields enter the login credentials of the Autonomous database user who is linked to the Oracle Data Transforms instance created earlier.
8. After entering the login credentials, click **Sign In**.
   The **Oracle Database Actions** page appears.

9. On the **Oracle Database Actions** page, click the Hamburger icon present on the top left corner.
10. From the left pane click **Data Tools** and select **Data Transforms**.
The **Oracle Data Transforms** login page appears. Upon successful login, **Oracle Data Transforms** page appears allowing you to perform all the data transformations.

16.5 Access Oracle Data Transforms From Data Studio

APPLIES TO: Data Transforms that is part of the suite of data tools built into Oracle Autonomous Database.

Data Transforms combines all the elements of data integration - data movement, data synchronization, data quality, and data management to ensure that information is timely, accurate, and consistent across complex systems.

To use the Data Transforms tool you must access Database Actions as the ADMIN user or have the DATA_TRANSFORM_USER role assigned. See Manage User Profiles with Autonomous Database for information on granting roles.

To access the Data Transforms tool:

1. Login as an ADMIN user or a user with the DATA_TRANSFORM_USER role assigned.
2. Click **Data Transforms** in the Database Actions page, or click the Selector icon and select **Data Transforms** from the Data Studio menu in the navigation pane.

> **Note:**
>
> Data Transforms is also available as a separate listing on OCI Marketplace. If you have already registered a Data Transforms instance from OCI Marketplace with the Autonomous Database, the Data Transforms card on the Database Actions page will continue to take you to your Marketplace instance. If you wish to use the embedded Data Transforms, then you must unregister the Marketplace instance. See Unregister the ODI Instance from Autonomous Database.

3. When you login to the Data Transforms tool for the first time, you need to provide the database user credentials to sign in.
It takes approximately 1-2 minutes for the Data Transforms service to start. After the service starts, the Data Transforms home page opens.
4. From the left pane of the Home page, click the **Connections** tab to view the newly created Autonomous Database connection.
4. Click the **Actions** icon next to the connection and select **Edit**.
5. In the **Update Connection** page, enter the database username and password to use the connection. Click **Update** to save the changes.

16.6 Enable Access to Private Data Sources from Autonomous Database

APPLIES TO: Data Transforms that is part of the suite of data tools built into Oracle Autonomous Database.

The Autonomous Database from where you are accessing Data Transforms should be configured to use a private endpoint to be able to communicate with private database sources. Otherwise when you try to create and test such a connection, you might get a "failed to connect" error.

If your Autonomous Database is configured to use a Private Endpoint, then you can only access private data sources from clients in the same Virtual Cloud Network (VCN). See Configuring Network Access with Private Endpoints for detailed instructions.

After you have configured network access using private endpoints, you need to do the following additional configurations for Oracle Data Transforms to communicate with private data sources. This chapter includes the following topics:

- Configuring DNS and Traffic Management
- Configuring RCE Proxy on the Private Endpoint

Configuring DNS and Traffic Management

You need to set up a fully qualified domain name (FQDN) and specify this FQDN in the connection host field when you create the connection. Refer to https://docs.oracle.com/en-us/iaas/Content/DNS/Concepts/views.htm for more information.

The VCN DNS resolver should know about the FQDN and be able to resolve it to a Type A IP address.

If you are not able to test a connection in Data Transforms on an Autonomous Database that is configured to use private endpoints, then do the following:

- Check whether the query to set the `ROUTE_OUTBOUND_CONNECTIONS` database property to `PRIVATE_ENDPOINT` is run. See Enhanced Security for Outbound Connections with Private Endpoints for more information.
- Make sure that the FQDN is used as the host so that the framework resolves the FQDN to the required IP address.
  Ensure that the data source VM has the FQDN set up with the required security rules.

If test connection fails even after using an FQDN, then do the following:

- From the OCI services menu, click **DNS Management** under **Networking**. Configure private DNS zones views and resolvers.
- Create and publish a record using the FQDN and the private IP (Type A) of the target database.

> **Note:**
>
> Do not use `*.oraclecloud.com` as the domain name when you set up the FQDN because it is a reserved domain name.

- Retry test connection.
  The test connection operation should complete successfully.

Configuring RCE Proxy on the Private Endpoint

For Autonomous Databases that are configured to use a private endpoint, the Autonomous Database service supports RCE to access resources in the customer subnet through a private route.

If you encounter a "failed to connect" error when you try to test a connection, contact your database administrator to check whether the database is RCE-enabled.

Note that RCE proxy can forward the reverse traffic only to the default OCI DNS resolver. If there is any overridden resolver, then connectivity will fail.

16.7 Troubleshoot Mismatch with Server Cert DN Error

APPLIES TO: Data Transforms that is part of the suite of data tools built into Oracle Autonomous Database.

For a connection on an Autonomous Database that is configured to use private endpoints, you must specify the distinguished name (DN) of the database server in the JDBC URL so that the Oracle Connection Manager (CMAN) can accept the request. Otherwise when you try to create and test such a connection, you might get a "Mismatch with Server Cert DN" error.

First, specify `ssl_server_dn_match=yes` in the JDBC URL for the CMAN to accept the request. Next, use the `ssl_server_cert_dn` parameter to specify the DN of the database server. Note that the order in which the keys are placed in `ssl_server_cert_dn` is important. The sequence should be `CN`, `O`, `L`, `ST`, `C`.

For example:

```
jdbc:oracle:thin:@(description= (retry_count=20)(retry_delay=3)
(address=(protocol=tcps)(port=1521)(host=xxxxx.adb.us-phoenix-1.oraclevcn.com))
(connect_data=(service_name=xxxxx.adb.oraclecloud.com))
(security=(ssl_server_dn_match=yes)(ssl_server_cert_dn="CN=adwc.uscom-
east-1.oraclecloud.com, O=Oracle Corporation, L=Redwood City, ST=California,
C=US")))
```

To get the values for `ssl_server_cert_dn`:

1. Login to the VM that is in the same subnet where your Autonomous Database lies.
2. Type the following command:
   `openssl s_client -connect xxxxx.adb.us-phoenix-1.oraclevcn.com:1521 -showcerts`

   You will find the values for the following under Server Certificate details:

   CN, O, L, ST, C

   For example, "`CN=adwc.uscom-east-1.oraclecloud.com, O=Oracle Corporation, L=Redwood City, ST=California, C=US`"

17
Reference

Get information about Oracle Data Tranforms such as the installation location, log file information as well as instructions for resetting the schema user password, using Run Commands in your Data Transforms instance, and increasing the memory of the ODI Agent.

Topics

- Terminology Information
  There are similarities in the functionality between Oracle Data Integrator and Oracle Data Transforms, however, the terminology is different.
- Installation Location
  You may need this information if you want to change any aspect of the installation.
- Log Files Location
  This topic lists the log files that help you to keep a track record on all the events happening in your Oracle Data Transforms instance on Oracle Cloud Marketplace.
- Reset the Schema User Password
  The schema password expires every 180 days from the repository creation date or from the last password reset date. When you try to connect to such an existing repository the Data Transforms instance provisioning fails. You need to reset the password in the Autonomous Database as well the local wallet.
- Use the OCI Console to Run Commands in the Data Transforms Instance
  When the Oracle Data Transforms repository provisioning process completes, certain commands are available for use in the compute instance Run command feature.
- Increase the Memory of ODI Agent
- Backup and Restore Data Transforms Repositories
  You can move repositories from one database instance to another or backup and restore your existing repositories. Note that this is applicable only for Oracle repositories.

17.1 Terminology Information

There are similarities in the functionality between Oracle Data Integrator and Oracle Data Transforms, however, the terminology is different.

The differences in the terminology used in Oracle Data Integrator Studio and Oracle Data Transforms are:

Oracle Data Integrator Studio	Oracle Data Transforms	Functions
Data Server	Connection	Represents the physical object where data is stored. Examples include Database Instances, File Servers, and Cloud Application Instances. The properties contain all information required to connect and access data.

Oracle Data Integrator Studio	Oracle Data Transforms	Functions
Technology	Connection Type	Examples include Oracle, IBM DB2, Oracle Netsuite, Oracle Object Storage. For the list of supported technologies, see Supported Connection Types.
Physical Schema	Schema	Examples include Database Schema, File server Folder, Object Storage Bucket.
Data Store	Data Entity	A Tabular representation of a data structure. Examples include Database Tables, Files.
Reverse Engineering	Import Data Entity	The process of obtaining the metadata for a filtered set of objects in a Schema. Examples include Table Definitions from an Oracle Schema, VOs from a Cloud Applications Offering.
Model	Data Entity	The container for the imported objects. The properties define the rules for importing (how to filter and so on).
Project	Project	Container for the Transformation Design components (Mappings, Packages, Jobs). This doesn't include Data Servers, Schemas, Data Stores and Folders (which are shared between Projects)
Mapping	Data Flow	The Transformation Design. Defines how data flows from one Data Store to another, and how it is transformed.
Package	Workflow	Defines the sequence in which Mappings will be executed, together with what happens of failure conditions.
Job / Session	Job	The execution of a Package, Mapping or Reverse Engineer.
Schedule	Schedule	The rules for when a session executes.
Resource	Resource	Something you can schedule (a mapping or package).

17.2 Installation Location

You may need this information if you want to change any aspect of the installation.

The following installation location is used by the Oracle Data Transforms image.

Table 17-1 Installation Location

Area	Location on Server
MW_HOME	/u01/oracle/transforms_home

17.3 Log Files Location

This topic lists the log files that help you to keep a track record on all the events happening in your Oracle Data Transforms instance on Oracle Cloud Marketplace.

Log Files	Location
Agent application Logs	/u01/oracle/transforms_home/logs/app.log
Rest application Logs	/u01/oracle/transforms_home/logs/odi_adp_rest_txt.log
Instance provisioning Logs (used only during provisioning)	/u01/oracle/transforms_home/logs/odi_lean_install.log
Repo creation Logs (used only during provisioning)	/u01/oracle/transforms_home/logs/repoCreation.log
Configuration Logs (used only during provisioning)	/u01/oracle/transforms_home/logs/transformsStartup.log

17.4 Reset the Schema User Password

The schema password expires every 180 days from the repository creation date or from the last password reset date. When you try to connect to such an existing repository the Data Transforms instance provisioning fails. You need to reset the password in the Autonomous Database as well the local wallet.

To reset the schema user password:

1. Login to the Autonomous Database that was used for creating the ODI repository.
2. Run the following command to reset the password for the user:

   `alter user username identified by newpassword`
3. Connect to the compute instance that is running Oracle Data Transforms.
4. Navigate to the Data Transforms desktop and double click the `updatepassword.desktop` shortcut icon. In the terminal window that appears, enter the new password.

   This updates the password in the local wallet and restarts the jetty server.

> **Note:**
>
> Skip steps 3 and 4 if the instance that you want to connect to is no longer available and you are creating a new instance using the existing repository.

17.5 Use the OCI Console to Run Commands in the Data Transforms Instance

When the Oracle Data Transforms repository provisioning process completes, certain commands are available for use in the compute instance Run command feature.

You can use these commands as templates to create new commands that you can run without having to login into the instance.

Name	Command	Description
restartCommand	`sudo systemctl restart jettyserver.service`	Restarts the jetty server.
archiveCommand	`cp -R /u01/oracle/transforms_home/logs /tmp`	Archives the log files to the local temporary folder.
listLogs	`ls /u01/oracle/transforms_home/logs`	Lists the log files.
displayStartupLog	`cat /u01/oracle/transforms_home/logs/transformsStartup.log`	Displays the configuration logs. This information is useful for troubleshooting any configuration issues.

To use the run command feature, you have to setup the following policy:

```
Allow dynamic-group odi_group  to  manage instance-family  in compartment odi
Allow dynamic-group odi_group  to use instance-agent-command-family in
compartment odi
Allow dynamic-group odi_group to use instance-agent-command-execution-family
in compartment odi where request.instance.id=target.instance.id
```

To use the pre-installed scripts to create commands to run on an instance:

1. On the OCI console, open the navigation menu present on the top left corner and click **Compute**.
2. Under **Compute**, click **Instances**.
3. Click the instance that you want to run the command on.
4. Under **Resources**, click **Run command**. The Run command screen lists the commands that you can use as a template to create and execute commands.
5. Find the command in the list, click the Actions menu, and then click **View command details**.
6. On the Command details page, copy the command from the **Command Content** section and close the page.
7. Click **Create command**.
8. Enter a name for the command.
9. In the **Add script** section, select **Paste script** and paste the copied command in the box.
10. In the **Output type** section, select the location to save the output of the command.
11. Click **Create command**.

12. Wait for the completion of the command. See the **Execution Status** column to monitor whether the command has run successfully.

If the command output was saved to an Object Storage location, either download the response object from the bucket where it was saved or navigate to the Object Storage pre-authenticated request URL. If the command output was saved as a plain text file, find the command in the list, click the Actions menu, and then click **View command details** to view the output.

17.6 Increase the Memory of ODI Agent

APPLIES TO: Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

If the data that you are loading from the source schema is huge, then you may want to increase the memory of the ODI Agent to avoid OutOfMemory exception errors.

To increase the memory of the ODI Agent:

1. Edit the `/u01/oracle/transforms_home/common/scripts/jettyServer.sh` file.
2. Add the `java -Xms1024m -Xmx4096m` parameter.
3. Restart the jetty server. Log in as OPC user and execute the following commands:

```
ssh -i <path to id_rsa> opc@<Instance IP>
sudo su
systemctl stop|start jettyserver.service
exit
```

17.7 Backup and Restore Data Transforms Repositories

You can move repositories from one database instance to another or backup and restore your existing repositories. Note that this is applicable only for Oracle repositories.

The topics within this section describe how you can:

- Use SQL Developer to Export and Import Data Transforms Repositories
  Use the SQL Developer Data Pump feature to export the source schema into a dump file and then import the dump file into another repository.
- Switch to the Imported Repository
  To switch to the imported repository, you need to update the `odi-setup.properties` and `repository.properties` files with information related to the imported repository.
- Update the Imported Repository using DataTransformsPostImport.sh
  Use the `DataTransformsPostImport.sh` utility to make the necessary updates to the imported repository.

17.7.1 Use SQL Developer to Export and Import Data Transforms Repositories

Use the SQL Developer Data Pump feature to export the source schema into a dump file and then import the dump file into another repository.

This topic has the following sections:

- Export Source Schema

- Copy the Dump File to Oracle Cloud Object Storage
- Create the Credential Object to Connect to Object Storage
- Create the Pre-Authenticated URL
- Import the Data Transforms Repository dump
- Stop the jetty server

Follow the below steps to import the ODI repository dump:

Export Source Schema

Run the Export Wizard from SQL Developer to export the source schema into a dump file, which you will later import into the target repository.

To export the Data Transforms repository:

1. Start SQL Developer.
2. Define a connection to connect to the source repository. Set up proxy, if needed.
3. Use this connection to open the DBA panel.
4. Add the newly created connection.
5. Expand the **Data Pump** node, right-click **Export Jobs** and select **Data Pump Export Wizard**. This opens the Export Wizard.
6. In the **Source** screen, provide the following information:
   - **Data or DDL**: Select **Data and DDL** from the drop-down list.
   - **Types to be Exported in Data Pump**: Select **Schemas**.
7. Click **Next**.
8. In the **Schemas** screen select the source schema that you want to export. Click **Next**.
9. Leave the default settings in the **Filter** and **Table Data** screens as is and click **Next**.
10. In the **Options** screen select the Enable Logging checkbox to enable logging. This generates an export log file, which will be saved in the DATA_PUMP_DIR directory. You can either leave the default log file name as is or change it. Click **Next**.
11. In the **Output Options** screen, the Choose Output Files section displays a list of output files with the .DMP extension. Select the file that you want to export. This dump file will also be saved in the DATA_PUMP_DIR directory. Select the **Append Timestamp to Dump, Log and Job names** radio button and click **Next**.
12. In the **Job Schedule** screen, specify information about the job such as the Job Name and Job Description. Select the required scheduling options or accept the default settings to run the job immediately. Click **Next**.
13. In the **Summary** screen, review the displayed information.
14. Click **Finish** to start the export process.
    A new export job will be added in the DBA window. Select the job from the **Export Jobs** node to check the status. You can also see the execution log in the lower-right window.

    When the job completes successfully, the Export Log and the dump file can be found in the **DATA_PUMP_DIR** directory.

Copy the Dump File to Oracle Cloud Object Storage

You need to copy the exported dump file to Oracle Cloud Object Storage so that you can access it from the target repository. You can use the Object Storage of either the source or

target tenant. This will allow you to import the `.dmp` file from any tenant, as long as you have the corresponding authentication token in order to access its Object Storage. To get the exported file, use the PL/SQL `PUT_OBJECT` function.

- Open a SQL worksheet connected to the source database where the data pump export happened.
- Run the following function in SQL Developer:

```
BEGIN
   DBMS_CLOUD.PUT_OBJECT(credential_name => 'ODI',
     object_uri => ' https://objectstorage.<your-
region>.oraclecloud.com/n/<your-namespace>/b/<your-bucket>/o/
SourceSchemaName.dmp ',
     directory_name => 'DATA_PUMP_DIR',
     file_name =>'NameOfDumpFile.DMP');
END;
```

  Here

  – The credential name "ODI" should have already been created in the source database. Check all the existing credentials from USER_CREDENTIALS.
  – The value for `object_uri` can be obtained from the properties of the target bucket on Object Storage.

Create the Credential Object to Connect to Object Storage

Create a credential object on the target repository to access the Object Storage to import the dump file. The first step is to create an authentication token in Oracle Object Storage. See Getting an Auth Token for information about how to generate the auth token. Copy and save the generated token. You will use this token to create the credential object to connect to the Object Store.

To create the credential object, execute the following script:

```
BEGIN

    DBMS_CLOUD.CREATE_CREDENTIAL(

        credential_name => 'Credential name',
        username => 'oci username',
        password => '<Enter the Authentication token here>');
 END;
```

Create the Pre-Authenticated URL

To access the DMP files from Object Storage you need to create pre-authenticated requests for each file that will be used for the import. When you create a Pre-authenticated Request for an object in Oracle Object Storage, you get the URL to the object. Copy and save this URL. You need to provide this URL in the Data Pump Import Wizard to import the dump file. See Creating a Pre-Authenticated Request for information about how to create the URL.

Import the Data Transforms Repository dump

Run the Import Wizard from SQL Developer to import the dump file into the target repository.

To import the repository dump:

1. Start SQL Developer.
2. Define a connection to connect to the target repository.
3. Use this connection to open the DBA panel.
4. Add the newly created connection.
5. Expand the **Data Pump** node, right-click **Import Jobs** and select **Data Pump Import Wizard**. This opens the Import Wizard.
6. In the **Type** screen, provide the following information:
   - **Data or DDL**: Select **Data and DDL** from the drop-down list.
   - **Type of Import**: Select **Schemas**.
   - In **Choose Input Files** set the location of the DMP file(s) by selecting the Credential that you created in Create the Credential Object to Connect to Object Storage and provide the pre-authenticated URI that you created in Create the Pre-Authenticated URL.
7. If the credential connection is successful, the input file is scanned, and the **Filter** screen displays all the available DMP schemas. Select the source schema(s) that you would like to import into the target repository. Click **Next**.
8. In the **Remapping** screen map the Source and Destination schemas. Click **Add Row**. The source schema will be automatically displayed. Enter the destination schema. The source schema will be imported to this empty schema. Click **Next**.
9. In the **Options** screen select the Enable Logging checkbox to enable logging. This generates an import log file, which will be saved in the DATA_PUMP_DIR directory. You can either leave the default log file name as is or change it. Click **Next**.
10. In the **Schedule** screen, specify information about the job such as the Job Name and Job Description. Select the required scheduling options or accept the default settings to run the job immediately. Click **Next**.
11. In the **Summary** screen, review the displayed information.
12. Click **Finish** to start the import process.
    A new import job will be added in the DBA window. Select the job from the **Import Jobs** node to check the status. You can also see the execution log in the lower-right window.

    When the job completes successfully, the Import Log and the dump file can be found in the **DATA_PUMP_DIR** directory.

Stop the jetty server

After you have imported the repository, you must stop the jetty server. To stop the jetty server, login as OPC user and execute the following commands:

```
ssh -i <path to id_rsa> opc@<Instance IP>
sudo su
systemctl stop jettyserver.service
exit
```

17.7.2 Switch to the Imported Repository

To switch to the imported repository, you need to update the `odi-setup.properties` and `repository.properties` files with information related to the imported repository.

To switch to the imported repository:

1. Create the `odi-setup.properties` file in `/u01/oracle/transforms_home/common/scripts` and add the following properties. If the file already exists, replace the existing content of the file with these details.

```
dbTech=ADB
rcuCreationMode=false
odiSchemaPassword=<valid password>
odiSchemaUser=<odi schema username>
odiSupervisorPassword=<odi SUPERVISOR password>
walletZipLoc=<path_to_zipped_wallet>
workRepoName=<WORK REPO NAME>
adwInstancePassword= <adw Instance password>
```

   Here,

   - `workRepoName=<WORK REPO NAME>` is an optional property but you may have to configure this property if your default work repository name is not `WORKREP`.
   - `adwInstancePassword= <adw Instance password>` is an optional property but configure this property only when you have used OPTACH for applying a patch on your ODI instance and wish to run Upgrade Assistant (UA) using the configuration script `odiMPConfiguration.py`.

2. Create the `repository.properties` file in `/u01/oracle/transforms_home/common/scripts` and add the following properties. If the file already exists, replace the existing content of the file with these details.

```
masterReposDriver=oracle.jdbc.OracleDriver
masterReposUrl=jdbc\:oracle\:thin\:@<new work repository name>_high?
TNS_ADMIN\=/u01/oracle/transforms_home/wallets/wallet_<wallet file name>
workReposName=<WORK REPO NAME>
masterReposUser=<odi schema username>
```

17.7.3 Update the Imported Repository using DataTransformsPostImport.sh

Use the `DataTransformsPostImport.sh` utility to make the necessary updates to the imported repository.

Before you run the utility make sure that the `repository.properties` file has the correct information that points to the imported master repository. See Switch to the Imported Repository for more information.

The `DataTransformsPostImport.sh` file is located in `/u01/oracle/transforms_home/common/scripts`. If the script is not found, create the file using the following properties:

```
#!/bin/sh

export install_home=/u01/oracle/transforms_home

CLASSPATH=.:$install_home/apps/*:$install_home/deploy/jettydeploy/
APPADMINREST/webapp/WEB-INF/lib/ojdbc8-19.3.0.0.jar:$install_home/deploy/
jettydeploy/APPADMINREST/webapp/WEB-INF/lib/
oraclepki-19.3.0.0.jar:$install_home/deploy/jettydeploy/APPADMINREST/webapp/
WEB-INF/lib/osdt_cert-19.3.0.0.jar:$install_home/deploy/jettydeploy/
APPADMINREST/webapp/WEB-INF/lib/osdt_core-19.3.0.0.jar
export CLASSPATH
```

```
/u01/oracle/jdk1.8.0_211/bin/java -cp $CLASSPATH
oracle.odi.setup.util.mp.DataTransformsPostImport

exec $SHELL
```

After you have successfully run the utility, you must start the jetty server. To start the jetty server, login as OPC user and execute the following commands:

```
ssh -i <path to id_rsa> opc@<Instance IP>
sudo su
systemctl start jettyserver.service
exit
```

Connect to Data Transforms and verify whether the repository has been successfully imported.

18
Python API for Oracle Data Transforms

Oracle Data Transforms provides a powerful, easy-to-use web-based user interface to manage connections, data loads, ELT/ETL pipelines, workflows, and schedules. Data Transforms Python SDK enables developers to create and manage the same using the comprehensive suite of Python APIs.

Topics

18.1 SDK Basics

Before getting started with the Python SDK, it is important to understand the basic concepts of Data Transforms.

Deployment

Represents an instance where Data Transforms is running.

Data Transforms can be deployed from three places.

- OCI Marketplace
- ADB-S
- OCI GoldenGate Service.

Accessing the deployment varies based on the type of deployment. *Data Transforms Python SDK provides unified access for all of these deployment types.*

Deployment Config

Includes the configuration parameters for accessing the Deployment. The Deployment Config leverages Python's configparser with customised parameter(s) for Data Transforms.

- **Structure of the deployment configuration**

- The *deployment.config* file is where all the deployments are configured *(default implementation)*. This file captures one or more deployments with their connection parameters.
  - Every deployment is identified by a *unique name*, represented as *[name_of_the_deployment]*.
  - The deployment file **must have one ACTIVE deployment**, identified with *[ACTIVE]* entry. If `deployment.config` does not have any active configuration available, the API execution(s) will be skipped.
  - Active Deployment is identified by a *unique name* as below:

    ```
    [ACTIVE]
    deployment=<deployment-name>
    ```

- *Marketplace based deployment:*

  ```
  [mp_deployment]
  xforms_ip=<ip-address>
  xforms_user=<user>
  ```

  where
  - `<ip-address>` is obtained from Public IP of the compute instance where Data Transforms was provisioned from OCI Marketplace.
  - `xforms_user` is the login user created for accessing Data Transforms.
- *Autonomous Database based deployment:*

  ```
  [name_of_deployment]
  XFORMS_URL=<data-transforms-url>
  xforms_user=<user>
  tenancy_ocid=<tenancy_ocid>
  adw_name=<name of the ADW instance>
  adw_ocid=<ADW ocid>
  ```

  where
  - `<data-transforms-url>` is the URL of your Data Transforms instance.
    To get the URL:
    1. From the Oracle Cloud Infrastructure left navigation menu click **Oracle Database**, select **Autonomous Database**, select the Autonomous Database instance from the list, and navigate to the Autonomous Database details page.
    2. Select the **Tool configuration** tab.
    3. Copy the URL from the Public access URL text box **excluding the trailing slash (*/*)**.
       For example, `https://example.adb.us-phoenix-1.oraclecloudapps.com/odi`
  - `<user>` is the login user created for accessing Data Transforms.
  - `<tenancy_ocid>` is the tenancy OCID from the OCI Console.
  - `<name of the ADW instance>` is the name of the Autonomous Database instance where the compute node will be built.

You will find the database name under General Information in the Autonomous Database details page.

  * `<ADW ocid>` is the OCID of the Autonomous Database.

- *OCI GoldenGate based deployments:*

```
[name_of_deployent]
XFORMS_URL=<data-transforms-url>
xforms_user=<user>
```

- *Sample Deployment Config file:*

```
[ACTIVE]
deployment=mydev_env

[my_dev_env]
xforms_ip=127.0.0.1
xforms_user= user

[abds_env]
XFORMS_URL=https://my-adbs-transforms-url
xforms_user=user
tenancy_ocid=ocid1.tenancy.oc1..tenancy_ocid
adw_name=mydw
adw_ocid=ocid1.autonomousdatabase.oc1.phx.adw_ocid

[qa_env]
xforms_ip=127.0.0.2
xforms_user=user

[scott_env]
xforms_ip=127.1.0.1
xforms_user=scott_env_user
```

Workbench

A dedicated workspace for the developer. A directory where all the Python scripts are maintained. The workbench usually has the following:

- A `deployment.config` file.
- Python scripts for connections, data entities, projects.
- Artifacts in the respective project folder.

> **Note:**
>
> To generate detailed class level API documentation run the following command:
> `python -m pydoc -w <class-name>`
>
> For example,
>
> `python -m pydoc -w datatransforms.dataflow`

18.2 Installation and Setup

Setup assumes the Python environment is already installed and available.

Installation

- *Prepare the environment:*
  The steps below are provided for a standard venv of Python. If you are using other tools (for example, Conda) that might need additional third party software, refer to their instructions on how to prepare the environment.

  For Linux and Mac based systems:

  ```
  cd <your work directory>
  python3 -m venv '.venv'
  source '.venv/bin/activate'
  ```

  For Windows based systems:

  ```
  cd <your work directory>
  python3 -m venv 'venv'
  .\venv\bin\activate or activate.bat (based on your python3 installation)
  ```

- *Installation is done using the pip3 command as below:*

  ```
  pip3 install oracle-data-studio
  ```

Update `PYTHONPATH` environment variable

Once the package installation is complete, update `PYTHONPATH` with the site-packages location.

For Linux and Mac based systems:

```
export PYTHONPATH=./.venv/lib/python<your-py-version>/site-packages/
datatransforms:$PYTHONPATH
```

For Windows based systems:

```
export PYTHONPATH=.\venv\lib\python<your-py-version>\site-
packages\datatransforms;%PYTHONPATH%
```

18.3 SDK Usage: Common steps

The examples here provide information on how to get started with the API.

Common steps

Before using the SDK APIs, the below steps are common across the usage:

1. Import the workbench and workbench configuration class.
2. Load the default workbench configuration.
3. Initialise the workbench and connect the workbench to the active deployment.

```
from datatransforms.workbench import DataTransformsWorkbench,WorkbenchConfig
pswd = ... #securely fetch your password from keystore for deployment.
```

```
connect_params = WorkbenchConfig.get_workbench_config(pswd)
workbench = DataTransformsWorkbench()
workbench.connect_workbench(connect_params)
```

Managing Log Levels

Logging level is managed through the standard logging package of Python. A snippet showing how to update the log level is given below:

```
#import
import logging
#Change/Manage log level. INFO | DEBUG | ERROR
logging.getLogger().setLevel(logging.INFO)
```

18.4 Verify Deployment Connectivity

Create a directory for your project and prepare the Deployment Configuration File `deployment.config` inside your project folder.

```
[ACTIVE]
deployment=my_env

[my_env]
xforms_ip=<IP address of Data Transforms instance>
xforms_user= <user>

[my_adbs_env]
XFORMS_URL=<data transforms url>
xforms_user=<user>
tenancy_ocid=<tenancy OCID>
adw_name=<adw name given in OCI>
adw_ocid=<adw ocid>
```

> **Tip:**
>
> The above *deployment.config* connects to *my_env* deployment, which is marked ACTIVE.

Create `about.py` with the following code snippet:

```
from datatransforms.workbench import DataTransformsWorkbench,WorkbenchConfig
import logging

logging.getLogger().setLevel(logging.ERROR)
pswd = ... #securely fetch your password from keystore for deployment.
connect_params = WorkbenchConfig.get_workbench_config(pswd)
workbench = DataTransformsWorkbench()
workbench = DataTransformsWorkbench()
workbench.connect_workbench(connect_params)
```

```
workbench.print_about_string()
```

Directory structure of your project:

```
<ProjectDirectory>
|-deployment.config
|-about.py
```

Run `about.py` using your IDE or CLI. CLI based run is given below:

```
ProjectDirectory$ python3 about.py
```

Ouput produced from `about.py` gives all the details of the active Data Transforms instance:

```
Oracle Data Transforms - Deployment details
------------------------------------------------------------------------
repositoryVersion : <repo-vesion>
releaseName : <mp_listing_name>
Current Time in Deployment Instance : <current-time>
Deployment Type : <ADBS|Marketplac>
Deployment Instance : <REST URL of Data Transforms>
------------------------------------------------------------------------
```

18.5 Create Connections

Connections capture information on the systems where data is stored. The connections could be databases, cloud storage, applications, or services from where data is extracted or loaded.

This example illustrates a script that creates two Oracle connections in Data Transforms:

```
from datatransforms.workbench import DataTransformsWorkbench,WorkbenchConfig
from datatransforms.connection import Connection
from datatransforms.connection_types import  ConnectionTypes,
ConnectionTypeDrivers

pswd="<your deployment pswd from secret store>"
connect_params = WorkbenchConfig.get_workbench_config(pswd)
workbench = DataTransformsWorkbench()
workbench.connect_workbench(connect_params)

src_connection = Connection()\
    .connection_name("Demo Source Data")\
    .with_credentials("admin",Connection.encode_pwd("password goes here"))\
    .using_driver(ConnectionTypeDrivers.ORACLE)\
    .usingWallet("/path/of/the/source_dbwallet_file.zip")\
    .property("serviceName","your_adw_service")
src_connection.technology=ConnectionTypes.ORACLE.value

workbench.save_connection(src_connection)


dw_connection = Connection()\
```

```
    .connection_name("Demo Target Data")\
    .with_credentials("admin",Connection.encode_pwd("password goes here"))\
    .using_driver(ConnectionTypeDrivers.ORACLE)\
    .usingWallet("/path/of/the/targt_dbwallet_file.zip")\
    .property("serviceName","datatransformsdemos_high")

dw_connection.technology=ConnectionTypes.ORACLE.value

workbench.save_connection(dw_connection)
```

> **WARNING:**
>
> Never check-in or manage the production code with plain text passwords.

18.6 Create Data Entities

Data entities represent the metadata of the source or target data, such as column/field name, data type, scale, etc. The data entitities can be created programmatically through the APIs or using the utility provided to discover the data entites and generate the Python code for the same.

This example shows a script that creates data entities in Data Transforms (less recommended):

```
from datatransforms.workbench import DataTransformsWorkbench,WorkbenchConfig
from datatransforms.dataentity import DataEntity

##Load active workbench from configuration
pswd="<your deployment pswd from secret store>"
connect_params = WorkbenchConfig.get_workbench_config(pswd)
workbench = DataTransformsWorkbench()
workbench.connect_workbench(connect_params)

channels_entity = DataEntity().from_connection("Demo Source
Data","SH").entity_name("CHANNELS")
channels_entity.add_column(name="CHANNEL_ID",position=1,dataType="NUMBER",data
TypeCode="NUMBER",length=0,scale=-127)
channels_entity.add_column(name="CHANNEL_DESC",position=2,dataType="VARCHAR2",
dataTypeCode="VARCHAR2",length=20,scale=0)
channels_entity.add_column(name="CHANNEL_CLASS",position=3,dataType="VARCHAR2"
,dataTypeCode="VARCHAR2",length=20,scale=0)
channels_entity.add_column(name="CHANNEL_CLASS_ID",position=4,dataType="NUMBER
",dataTypeCode="NUMBER",length=0,scale=-127)
channels_entity.add_column(name="CHANNEL_TOTAL",position=5,dataType="VARCHAR2"
,dataTypeCode="VARCHAR2",length=13,scale=0)
channels_entity.add_column(name="CHANNEL_TOTAL_ID",position=6,dataType="NUMBER
",dataTypeCode="NUMBER",length=0,scale=-127)
workbench.save_data_entity(channels_entity)
```

Using the utility to create the Python File (Preferred)

```
% python -m datatransforms.cli.generate_data_entities.py --connection "Demo
Source Data" --schema SH --live true

Oracle DataTransforms

Generating data entities using options ...
Namespace(log_level=None, connection='Demo Source Data', schema='SH',
live=True, matching=None)


INFO:root:Loading active work bench
INFO:root:Loading active deployment configuration
Data Transforms URL http://xxx.xxx.xxx.xx:9999/odi-rest
INFO:root:Data Entities will be generated in script file
demosourcedata_sh_entities.py
INFO:root:Fetching data entities using live connection
INFO:root:***No matching filter applied, 9 tables will be used to generate
data entities***
DataEntity Generation Complete
%
```

> **Note:**
>
> When `-live` is `true` the utility will go to the database to fetch the data entities. If it is `false`, it will fetch them from the Data Transforms Repository.

Notice that in the work area directory there is a new file called `demosourcedata_sh_entities.py`. View this file to see the Create Data Entity commands for all tables in the SH schema. Move this file to your project folder.

```
% mv demosourcedata_sh_entities.py MyProject
%
```

From your Work Folder, you can run this Python file to create the Data Entities.

```
% cd /Users/someuser/DT4D_Projects
% python3 MyProject/demosourcedata_sh_entities.py
Demo Source Data.SH.CHANNELS
Demo Source Data.SH.COSTS
Demo Source Data.SH.COUNTRIES
Demo Source Data.SH.CUSTOMERS
Demo Source Data.SH.PRODUCTS
Demo Source Data.SH.PROMOTIONS
Demo Source Data.SH.SALES
Demo Source Data.SH.SUPPLEMENTARY_DEMOGRAPHICS
Demo Source Data.SH.TIMES
%

If you like, you can connect to Data Transforms, navigate to the Data
```

```
Entities tab, select "Demo Source Data"
in the  connect filter to see the Data Entities
```

18.7 Load Data

Data Transforms allows you to directly load data from one system to another.

Create a file called `load_data.py` in your project directory with the following information.

> **Note:**
>
> You may need to change schema names based on your environment.

```
from datatransforms.workbench import DataTransformsWorkbench,WorkbenchConfig
from datatransforms.dataload import DataLoad
from datatransforms.project import Project
import logging

pswd="<your deployment pswd from secret store>"
connect_params = WorkbenchConfig.get_workbench_config(pswd)
workbench = DataTransformsWorkbench()
workbench.connect_workbench(connect_params)

prj = Project(name="MyProject")

###################################
#      Delete the Project         #
###################################
workbench.delete_project(prj)

###################################
#      Create a Project           #
###################################
project = Project("MyProject")
workbench.save_project(project)

###################################
#      Create a Data Load         #
###################################

dl = DataLoad("Load SH Data",project_name="MyProject")
dl.source("Demo Source Data.SH").\
recreate(table_name="COSTS").\
recreate(table_name="COUNTRIES").\
recreate(table_name="PRODUCTS").\
recreate(table_name="SUPPLEMENTARY_DEMOGRAPHICS").\
recreate(table_name="TIMES").\
recreate(table_name="CHANNELS").\
recreate(table_name="CUSTOMERS").\
recreate(table_name="PROMOTIONS").\
recreate(table_name="SALES").\
```

```
target("Demo Target Data.DEMO_TARGET")
dl.create_dataload()
```

You can navigate to the project in the Data Transforms UI and open the Data Load you just created and run it. Alternatively, you can add it to a Workflow and schedule it using the API.

18.8 Create Data Flows

You can create data pipelines using DataFlow APIs. A Data Flow is composed of sources (one or more), data transformation operators and a target.

You can customize the data loading strategies using the LoadOptions available for the specific connector.

Create the file `transform_data.py`.

```
from datatransforms.workbench import DataTransformsWorkbench,WorkbenchConfig
from datatransforms.dataflow import DataFlow
from datatransforms.dataentity import DataEntity
from datatransforms.dataflow_load_options import
OracleInsert,DataFlowIntegrationType
import logging

pswd="<your deployment pswd from secret store>"
connect_params = WorkbenchConfig.get_workbench_config(pswd)
workbench = DataTransformsWorkbench()
workbench.connect_workbench(connect_params)

############################################################
#      Create the Target Table for the Data Flow           #
############################################################
#pylint: disable=all
product_sales_entity = DataEntity().from_connection("Demo Target
Data","DEMO_TARGET").entity_name("PRODUCT_SALES")
product_sales_entity.add_column(name="CUST_ID",position=2,dataType="NUMBER",da
taTypeCode="NUMBER",length=38,scale=None)
product_sales_entity.add_column(name="TIME_ID",position=3,dataType="DATE",data
TypeCode="DATE",length=None,scale=None)
product_sales_entity.add_column(name="CHANNEL_ID",position=4,dataType="NUMBER"
,dataTypeCode="NUMBER",length=38,scale=None)
product_sales_entity.add_column(name="PROMO_ID",position=5,dataType="NUMBER",d
ataTypeCode="NUMBER",length=38,scale=None)
product_sales_entity.add_column(name="QUANTITY_SOLD",position=6,dataType="NUMB
ER",dataTypeCode="NUMBER",length=10,scale=2)
product_sales_entity.add_column(name="AMOUNT_SOLD",position=7,dataType="NUMBER
",dataTypeCode="NUMBER",length=10,scale=2)
product_sales_entity.add_column(name="PROD_ID",position=8,dataType="NUMBER",da
taTypeCode="NUMBER",length=6,scale=0)
product_sales_entity.add_column(name="PROD_NAME",position=9,dataType="VARCHAR2
",dataTypeCode="VARCHAR2",length=50,scale=None)
product_sales_entity.add_column(name="PROD_DESC",position=10,dataType="VARCHAR
2",dataTypeCode="VARCHAR2",length=4000,scale=None)
product_sales_entity.add_column(name="PROD_SUBCATEGORY",position=11,dataType="
VARCHAR2",dataTypeCode="VARCHAR2",length=50,scale=None)
product_sales_entity.add_column(name="PROD_SUBCATEGORY_ID",position=12,dataTyp
```

```
e="NUMBER",dataTypeCode="NUMBER",length=38,scale=None)
product_sales_entity.add_column(name="PROD_SUBCATEGORY_DESC",position=13,dataT
ype="VARCHAR2",dataTypeCode="VARCHAR2",length=2000,scale=None)
product_sales_entity.add_column(name="PROD_CATEGORY",position=14,dataType="VAR
CHAR2",dataTypeCode="VARCHAR2",length=50,scale=None)
product_sales_entity.add_column(name="PROD_CATEGORY_ID",position=15,dataType="
NUMBER",dataTypeCode="NUMBER",length=38,scale=None)
product_sales_entity.add_column(name="PROD_CATEGORY_DESC",position=16,dataType
="VARCHAR2",dataTypeCode="VARCHAR2",length=2000,scale=None)
product_sales_entity.add_column(name="PROD_WEIGHT_CLASS",position=17,dataType=
"NUMBER",dataTypeCode="NUMBER",length=3,scale=0)
product_sales_entity.add_column(name="PROD_UNIT_OF_MEASURE",position=18,dataTy
pe="VARCHAR2",dataTypeCode="VARCHAR2",length=20,scale=None)
product_sales_entity.add_column(name="PROD_PACK_SIZE",position=19,dataType="VA
RCHAR2",dataTypeCode="VARCHAR2",length=30,scale=None)
product_sales_entity.add_column(name="SUPPLIER_ID",position=20,dataType="NUMBE
R",dataTypeCode="NUMBER",length=6,scale=0)
product_sales_entity.add_column(name="PROD_STATUS",position=21,dataType="VARCH
AR2",dataTypeCode="VARCHAR2",length=20,scale=None)
product_sales_entity.add_column(name="PROD_LIST_PRICE",position=22,dataType="N
UMBER",dataTypeCode="NUMBER",length=8,scale=2)
product_sales_entity.add_column(name="PROD_MIN_PRICE",position=23,dataType="NU
MBER",dataTypeCode="NUMBER",length=8,scale=2)
product_sales_entity.add_column(name="PROD_TOTAL",position=24,dataType="VARCHA
R2",dataTypeCode="VARCHAR2",length=13,scale=None)
product_sales_entity.add_column(name="PROD_TOTAL_ID",position=25,dataType="NUM
BER",dataTypeCode="NUMBER",length=38,scale=None)
product_sales_entity.add_column(name="PROD_SRC_ID",position=26,dataType="NUMBE
R",dataTypeCode="NUMBER",length=38,scale=None)
product_sales_entity.add_column(name="PROD_EFF_FROM",position=27,dataType="DAT
E",dataTypeCode="DATE",length=None,scale=None)
product_sales_entity.add_column(name="PROD_EFF_TO",position=28,dataType="DATE"
,dataTypeCode="DATE",length=None,scale=None)
product_sales_entity.add_column(name="PROD_VALID",position=29,dataType="VARCHA
R2",dataTypeCode="VARCHAR2",length=1,scale=None)
workbench.save_data_entity(product_sales_entity)



###################################
#      Create a Data Flow         #
###################################

flow = DataFlow("Transform SH Data","MyProject")

#Define the Tables that will be used in the Data Flow
sales_data_entity=flow.use(connection_name="Demo Source
Data",schema_name="SH",data_entity_name="SALES")
products_data_entity=flow.use(connection_name="Demo Source
Data",schema_name="SH",data_entity_name="PRODUCTS")
productsales_data_entity=flow.use(connection_name="Demo Target
Data",schema_name="DEMO_TARGET",data_entity_name="PRODUCT_SALES")



#By Default, Source columns are mapped to target columns by matching name.
```

```
You can override column mappings
manual_column_mappings={
      "PRODUCT_SALES.PROD_NAME": "UPPER(PRODUCTS.PROD_NAME)"
}
override_column_mappings = {
    "column_mappings" : manual_column_mappings
}

OracleInsert.DROP_AND_CREATE_TARGET_TABLE=True
load_options=DataFlowIntegrationType.append()
load_options.update(OracleInsert.options())
load_options.update(override_column_mappings)


flow.from_source("SALES","Demo Source Data.SH.SALES").\
     from_source("PRODUCTS","Demo Source Data.SH.PRODUCTS").\
     join("Join","INNER","SALES.PROD_ID=PRODUCTS.PROD_ID").\
     filter_by("Filter","PRODUCTS.PROD_CATEGORY like 'Photo'").\
     load("PRODUCT_SALES","Demo Target
Data.DEMO_TARGET.PRODUCT_SALES",load_options)

flow.create()
```

18.9 Create Workflows

Create the file `orchestrate_jobs.py`.

```
from datatransforms.workbench import DataTransformsWorkbench,WorkbenchConfig
from datatransforms.workflow import Workflow,DataFlowStep,SqlStep,DataLoadStep

pswd="<your deployment pswd from secret store>"
connect_params = WorkbenchConfig.get_workbench_config(pswd)
workbench = DataTransformsWorkbench()
workbench.connect_workbench(connect_params)

###################################
#       Create a Worklow          #
###################################

#define steps in workflow , each step has name and its type
dls = DataLoadStep("Load SH Data","Load SH Data",firstStep=True)
dfs=DataFlowStep("Transform SH Data",data_flow_name="Transform SH
Data",data_flow_project="MyProject")
ss_success = SqlStep("Processing Success","Demo Target Data","select SYSDATE
from DUAL")
ss_failed = SqlStep("Processing Failed","Demo Target Data","select SYSDATE
from DUAL")

#define the execution order
dls.ok(dfs).ok(ss_success)
dls.nok(ss_failed)
dfs.nok(ss_failed)

wf = Workflow("Process SH Data",project="MyProject")
```

```
wf.add_execution_steps([dls,dfs,ss_success,ss_failed])
wf.create()
```

18.10 Schedule Jobs

You can create a schedule for when a job should run. This example shows how to schedule a job to run immediately.

```
import logging

from datatransforms.schedule import Schedule,SCHEDULE_STATUS_ACTIVE
from datatransforms.workbench import DataTransformsWorkbench,WorkbenchConfig

logging.getLogger().setLevel(logging.ERROR)

#use params from config , dont hardcode passwords. Used for unit test only
pswd="<your deployment pswd from secret store>"
connect_params = WorkbenchConfig.get_workbench_config(pswd)
workbench = DataTransformsWorkbench()
workbench.connect_workbench(connect_params)


#schedule DemoWorkflow2 , at the earliest possible. default next miute.
#due to network and latency it is recommended to use default;
# however it can be customized with asap option.
#NOT recommended to go less than 5 seconds unless instance connectivity is
fast (eg same network).
schedule = Schedule("test_schedule_immediate")\
    .workflow(project="MyProject",workflow_name="Process SH Data")\
    .immediate()\
    .schedule_status(SCHEDULE_STATUS_ACTIVE)
workbench.save_schedule(schedule)
```

18.11 Manage Credentials and Sensitive Fields

An approach for handling sensitive data in code (for example, passwords) is provided below.

> **Tip:**
>
> The approach is provided with emphasis on secure development practices to follow during development. It doesn't mandate a specific tool or technology.

Implementing Custom SecretUtility class

```
from datatransforms.secrets_util import SecretsStore

class MySecretManager(SecretsStore):

    ....
    #implement the password fetch/store contracts
```

```
    #if the custom class allows ONLY fetch passwords,
    #have exception thrown for store
```

This will be the default password manager for the modules while creating connections.

```
Connection con = Connection()
#fetches the password from the secret store implementation
con.set_password(MySecretManager().getPass("MyconnectionID"))
```