## Contents

### Preface
- Audience \( v \)
- Documentation Accessibility \( v \)
- Related Documents \( v \)
- Conventions \( vi \)

### 1 Getting Started with Autonomous Database
- About Oracle Cloud \( 1-1 \)
- Types of Cloud Accounts \( 1-1 \)
- Sign In to Your Account \( 1-2 \)
- Provision Autonomous Data Warehouse \( 1-2 \)

### 2 Connecting to Autonomous Database
- Connect with Built-in Oracle Database Actions \( 2-1 \)
  - About Database Actions (SQL Developer Web) \( 2-1 \)
  - Access Database Actions as ADMIN \( 2-2 \)
  - Use Database Actions to Query Data \( 2-2 \)
- Download Client Credentials (Wallets) \( 2-4 \)
- Connect Oracle SQL Developer with a Wallet (mTLS) \( 2-6 \)
- Predefined Database Service Names for Autonomous Data Warehouse \( 2-8 \)

### 3 Loading Data
- About Data Loading \( 3-1 \)
- Load Data with Oracle Database Actions \( 3-1 \)
- Load Data from Local Files with Oracle Database Actions \( 3-2 \)
  - Load Data into Existing Autonomous Database Table with Oracle Database Actions \( 3-3 \)
- Load Data from Files in the Cloud \( 3-6 \)
  - Create Credentials and Copy Data into an Existing Table \( 3-6 \)
  - Monitor and Troubleshoot Loads \( 3-8 \)
4  Querying External Data with Autonomous Database
   Query External Data 4-1
   Validate External Data 4-3
   View Logs for Data Validation 4-4

5  Working with Analytics and Visualization
   Using Oracle Analytics Desktop with Autonomous Data Warehouse 5-1
      Connect with Oracle Analytics Desktop 5-1
   Use Oracle Analytics Cloud with Autonomous Data Warehouse 5-2

6  Creating Web and Mobile Applications with Oracle Application Express
   About Oracle Application Express 6-1
   Access Oracle Application Express Administration Services 6-2
   Create Oracle Application Express Workspaces in Autonomous Data Warehouse 6-4
   Access Oracle Application Express App Builder 6-5
   Create Oracle Application Express Developer Accounts 6-5
   Use Web Services with Oracle Application Express 6-6
   Send Email from Oracle Application Express 6-8
   Restrictions and Limitations for Oracle Application Express with Autonomous Data Warehouse 6-10
Preface

This document describes how to manage, monitor, and use Oracle Autonomous Data Warehouse and provides references to related documentation.

Audience

This document is intended for Oracle Cloud users who want to manage and monitor Oracle Autonomous Data Warehouse.

This document is also intended for developers and end users who want to load and query data in Oracle Autonomous Data Warehouse.

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

Depending on the region and when you provisioned your database, and in some cases depending on your provisioning choice, the Oracle Database version for your Autonomous Database is either Oracle Database 19c or Oracle Database 21c.

If you have Oracle Database 19c, then many database concepts and features of this service are further documented here:

Oracle Database 19c

If you are using Always Free Autonomous Database with Oracle Database 21c, then many concepts and features of this service are further documented here:

Oracle Database 21c

For additional information, see these Oracle resources:

• Welcome to Oracle Cloud Infrastructure
• Oracle Cloud Infrastructure Object Storage
• Get Started Using Autonomous JSON Database
• GoldenGate Real-Time Data Replication in Cloud
Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td><strong>monospace</strong></td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
Getting Started with Autonomous Database

Provides an overview of the service and describes how to get started with Autonomous Data Warehouse.

Topics

• About Oracle Cloud
• Types of Cloud Accounts
• Sign In to Your Account
• Provision Autonomous Data Warehouse

About Oracle Cloud

Oracle Cloud is one of the few cloud providers that can offer a complete set of cloud services to meet all your enterprise computing needs.

Use Oracle Infrastructure as a Service (IaaS) offerings to quickly set up the virtual machines, storage, and networking capabilities you need to run just about any kind of workload. Your infrastructure is managed, hosted, and supported by Oracle.

Use Oracle Platform as a Service offerings to provision ready-to-use environments for your enterprise IT and development teams, so they can build and deploy applications, based on proven Oracle databases and application servers.

Use Oracle Software as a Service (SaaS) offerings to run your business from the Cloud. Oracle offers cloud-based solutions for Human Capital Management, Enterprise Resource Planning, Supply Chain Management, and many other applications, all managed, hosted, and supported by Oracle.

Types of Cloud Accounts

Oracle offers two types of Cloud Accounts: Free Tier Accounts and Oracle Cloud Paid Accounts.

• **Free Tier Accounts**: After you sign up for the free Oracle Cloud promotion or sign up for a paid account, you’ll get a welcome email. The email provides you with your cloud account details and sign in credentials.

  See Oracle Cloud Free Tier for more information.

• **Oracle Cloud Paid Accounts**: When your tenancy is provisioned, Oracle sends an email to the default administrator with the sign-in credentials and URL. This administrator can then create a user for each person who needs access to the Oracle Cloud. Check your email or contact your administrator for your credentials and account name.

  See Upgrade Your Free Oracle Cloud Promotion for more information.

See Request and Manage Free Oracle Cloud Promotions for more information.
Sign In to Your Account

If you don't have access to your Welcome email, you can sign in to your Cloud Account from the Oracle Cloud website.

Before You Begin

To sign in via the Oracle Cloud website, you must have:

- The name of your Cloud Account, if you are signing in to a Cloud Account with Identity Cloud Service. This is the Cloud Account name you have chosen during account signup. You'll find the Cloud Account name in your welcome email.
  OR
- Your data center and identity domain, if you are signing in to a Traditional Cloud Account
- The user name and password for your Cloud Account.

If you don't have this information, then click need help logging in and enter the email address associated with the Cloud Account. Oracle will send you an email with a summary of your account information.

Login to Oracle Cloud

To log in to your account from the Oracle Cloud website:

1. Point your browser to the following URL:
   http://oracle.com/

2. Click the ➔ View Accounts button in the upper right corner, and then click Sign in to Cloud underneath the Sign in to Cloud.

   Do NOT click the Sign-In button under ORACLE ACCOUNT, this will take you to Single Sign-On, not the Oracle Cloud

3. Enter the name of your Cloud Account. This is your Tenant that you received during sign up.

4. Enter your user name and password, and then click Sign In.

Once you successfully login, you will be presented with the Oracle Cloud homepage.

Provision Autonomous Data Warehouse

Follow these steps to provision a new Autonomous Data Warehouse instance using the Oracle Cloud Infrastructure Console.

Perform the following prerequisite steps as necessary:

- Open the Oracle Cloud Infrastructure Console by clicking the ➔ next to Oracle Cloud.
- From the Oracle Cloud Infrastructure left navigation menu click Oracle Database, and then click Autonomous Data Warehouse.
- Choose your region. See Switching Regions for information on switching regions and working in multiple regions.
• Choose your **Compartment**. See [Compartments](#) for information on using and managing compartments.

On the Autonomous Databases page, perform the following steps:

1. Click **Create Autonomous Database**.
2. Provide basic information for the Autonomous Database.
   - **Choose a compartment**. See [Compartments](#) for information on using and managing compartments.
   - **Display name** Specify a user-friendly description or other information that helps you easily identify the resource. The display name does not have to be unique.

   **Note:**
   After you create an Autonomous Database you cannot change the display name.

   - **Database name** Specify the database name; it must consist of letters and numbers only. The maximum length is 14 characters. The same database name cannot be used for multiple Autonomous Databases in the same tenancy in the same region.

3. Choose a workload type. Select the workload type for your database from the choices:
   - **Data Warehouse**
   - **Transaction Processing**
   - **JSON Database**
   - **APEX**

4. Choose a deployment type.
   - **Shared Infrastructure**
     Run Autonomous Database on shared Exadata infrastructure.
   - **Dedicated Infrastructure**
     Run Autonomous Database on dedicated Exadata infrastructure.

   Select **Shared Infrastructure** to create your instance on shared Exadata infrastructure.

   See [Create an Autonomous Data Warehouse Database on Dedicated Exadata Infrastructure](#) for steps to create your instance on dedicated Exadata infrastructure.

5. Configure the database.
   - **Always Free** Select to show Always Free configuration options.
     Always Free does not show when you select **JSON** workload type.
   - **Choose database version** Select the database version. The available database version is 19c.
     With Always Free selected, the available database versions are: 19c and 21c.
   - **OCPU Count** Specify the number of CPU cores for your database.
   - **Auto Scaling** By default auto scaling is enabled to allow the system to automatically use up to three times more CPU and IO resources to meet workload demand. If you do not want to use auto scaling then deselect this option to disable auto scaling.
     See [Use Auto Scaling](#) for more information.
6. Create administrator credentials. Set the password for the Autonomous Data Warehouse Admin user.
   - **Username** This is a read only field.
   - **Password** Set the password for the Autonomous Data Warehouse Admin user.
   - **Confirm password** Enter the same password again to confirm your new password.

   The password must meet the strong password complexity criteria based on Oracle Cloud security standards. For more information on the password complexity rules, see About User Passwords on Autonomous Database.

7. Choose network access

   **Note:**
   After you provision your Autonomous Database you can change the network access option you select for the instance.

   - **Secure access from everywhere**
     By default, secure connections are allowed from everywhere.

   - **Secure access from allowed IPs and VCNs only**
     This option restricts connections to the database according to the access control lists (ACLs) you specify. To add multiple ACLs for the Autonomous Database, click **+ Access Control Rule**.

     See Configure Access Control Lists When You Provision or Clone an Instance for more information.

   - **Private endpoint access only**
     This option assigns a private endpoint, private IP, and hostname to your database. Specifying this option allows traffic only from the VCN you specify; access to the database from all public IPs or VCNs is blocked. This allows you to define security rules, ingress/egress, at the Network Security Group (NSG) level and to control traffic to your Autonomous Database.

     See Configure Private Endpoints When You Provision or Clone an Instance for more information.

8. Choose a license type
   - **Bring Your Own License**
     My organization already owns Oracle database software licenses. Bring my existing database software licenses to the database cloud service (**details**).

   - **License Included**
     Subscribe to new database software licenses and the database cloud service.

9. (Optional) Provide up to 10 maintenance contacts
Click **Add Contact** and in the **Contact Email** field, enter a valid email address. To enter multiple **Contact Email** addresses, repeat the process to add up to 10 customer contact emails.

See View and Manage Customer Contacts for Operational Issues and Announcements for more information.

10. (Optional) Click Show Advanced Options to select advanced options.

   - **Encryption Key**
     - **Encryption using Oracle-managed keys**: By default Autonomous Database uses Oracle-managed encryption keys. Using Oracle-managed keys, Autonomous Database creates and manages the encryption keys that protect your data and Oracle handles rotation of the TDE master key.
     - **Encrypt using customer-managed keys**: If you select customer-managed keys, a master encryption key in the Oracle Cloud Infrastructure Vault is used to generate the TDE master key on Autonomous Database.
     
     See Use Customer-Managed Encryption Keys on Autonomous Database for more information.

   - **Maintenance**
     - **Patch level** By default the patch level is **Regular**. Select **Early** to configure the instance with the early patch level. Note: you cannot change the patch level after you provision an instance.
     
     See Set the Patch Level for more information.

   - **Tags**
     If you want to use Tags, enter the **TAG KEY** and **VALUE**. Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.
     
     See Tagging Overview for more information.

11. Click **Create Autonomous Database**.

    On the Oracle Cloud Infrastructure console the Lifecycle State shows **Provisioning** until the new database is available.
Connecting to Autonomous Database

Describes methods to securely connect to Autonomous Data Warehouse.

Topics
• Connect with Built-in Oracle Database Actions
• Download Client Credentials (Wallets)
• Connect Oracle SQL Developer with a Wallet (mTLS)
• Predefined Database Service Names for Autonomous Data Warehouse

Connect with Built-in Oracle Database Actions

You can access Database Actions from Autonomous Data Warehouse. Database Actions provides development tools, data tools, administration, and monitoring features for Autonomous Data Warehouse. Using Database Actions you can run SQL statements, queries, and scripts in a worksheet.

Topics
• About Database Actions (SQL Developer Web)
• Access Database Actions as ADMIN
• Use Database Actions to Query Data

About Database Actions (SQL Developer Web)

Database Actions provides a web-based interface with development, data tools, administration, and monitoring features in Autonomous Data Warehouse.

These are the main features of Database Actions:

• Development features:
  – Run SQL statements and scripts in the worksheet
  – Design Data Modeler diagrams using existing objects
  – Work with REST data services
  – Work with JSON data

• Data Tools features:
  – Data Load: Load or access data from local files, cloud storage, or from remote databases.
  – Data Insights: Discover anomalies, outliers, and hidden patterns in your data.
  – Catalog: Understand data dependencies and the impact of changes.
  – Business Models: Create business models for performance and analysis.
• Administration features:
  – Database user administration

See About Database Actions in Using Oracle Database Actions for more information.

Access Database Actions as ADMIN

Database Actions (also known as SQL Developer Web) is bundled with each Autonomous Database instance.

Database Actions runs in Oracle REST Data Services and access is provided through schema-based authentication. To use Database Actions, you must sign in as a database user whose schema is enabled for Database Actions. By default the ADMIN user is enabled to access Database Actions.

Note:
If your Autonomous Data Warehouse is configured to use a Private Endpoint, then you can only access Database Actions from clients in the same Virtual Cloud Network (VCN).

To access Database Actions from the Oracle Cloud Infrastructure Console:

1. On the Autonomous Database Details page click Database Actions.
2. On the Database Actions Launchpad, select a card.
   For example, click SQL to use a SQL Worksheet. On the SQL Worksheet you can use the Consumer Group drop-down list to select the consumer group to run your SQL or PL/SQL code. See Executing SQL Statements in the Worksheet Editor for more information.

Use Database Actions to Query Data

You can conveniently access the browser-based Database Actions from the Oracle Cloud Infrastructure console and use it to query the database.

This example shows you how to connect with Database Actions and perform a query on sample data sets provided out-of-the-box with Autonomous Database. You will run queries on sample data sets in Autonomous Database. Autonomous Database provides the Oracle Sales History sample schema and the Star Schema Benchmark (SSB) data set; these data sets are in the SH and SSB schemas, respectively.

You will run a basic query on the SSB data set which is a 1TB data set with one fact table with around 6 billion rows, and several dimension tables.

Access Database Actions from the Oracle Cloud Infrastructure Console:

1. On the Autonomous Database Details page click Database Actions.
2. To run a query on data, in Database Actions, under Development click SQL.
   This shows an SQL Worksheet.
3. To run a query on data, in the SQL worksheet, enter your query:
4. Copy the example below into the Database Actions worksheet.

```
SELECT /* low */ c_city, c_region, count(*) FROM ssb.customer c_low group
    BY c_region, c_city ORDER BY count(*);
```

5. Click **Run Statement** to execute the query.

This example shows the following output:
Download Client Credentials (Wallets)

To download client credentials you can use the Oracle Cloud Infrastructure Console or the Autonomous Database Service Console.

**Note:**

The password you provide when you download the wallet protects the downloaded Client Credentials wallet.

For commercial regions, the wallet password complexity for the password you supply requires the following:

- Minimum of 8 characters
- Minimum of 1 letter
- Minimum of 1 numeric character or 1 special character

For US Government regions, the wallet password complexity requires all of the following:

- Minimum of 15 characters
- Minimum of 1 lowercase letter
- Minimum of 1 uppercase letter
- Minimum of 1 numeric character
- Minimum of 1 special character

**To download client credentials from the Oracle Cloud Infrastructure Console:**

1. Navigate to the Autonomous Database details page.
2. Click **DB Connection**.
3. On the **Database Connection** page select the **Wallet Type**:
   - **Instance Wallet**: Wallet for a single database only; this provides a database-specific wallet.
   - **Regional Wallet**: Wallet for all Autonomous Databases for a given tenant and region (this includes all service instances that a cloud account owns).

**Note:**

Oracle recommends you provide a database-specific wallet, using **Instance Wallet**, to end users and for application use whenever possible. Regional wallets should only be used for administrative purposes that require potential access to all Autonomous Databases within a region.

4. Click **Download Wallet**.
5. In the **Download Wallet** dialog, enter a wallet password in the **Password** field and confirm the password in the **Confirm Password** field.

6. Click **Download** to save the client security credentials zip file.
   
   By default the filename is: `Wallet_databasename.zip`. You can save this file as any filename you want.
   
   You must protect this file to prevent unauthorized database access.

**To download client credentials from the Autonomous Data Warehouse Service Console:**

1. From the Service Console click the **Administration** link.

2. Click **Download Client Credentials (Wallet)**.

3. On the **Download Client Credentials (Wallet)** page, enter a wallet password in the **Password** field and confirm the password in the **Confirm Password** field.

4. Click **Download** to save the client security credentials zip file. By default the filename is: `Wallet_databasename.zip`. You can save this file as any filename you want. You must protect this file to prevent unauthorized database access.

---

**Note:**

When you use the Service Console to download a wallet there is no **Wallet Type** option on the **Download Client Credentials (Wallet)** page and you always download an instance wallet. If you need to download the regional wallet, use **DB Connection** on the Oracle Cloud Infrastructure console.

The zip file includes the following:

- **tnsnames.ora** and **sqlnet.ora**: Network configuration files storing connect descriptors and SQL*Net client side configuration.
- **cwallet.sso** and **ewallet.p12**: Auto-open SSO wallet and PKCS12 file. The PKCS12 file is protected by the wallet password provided while downloading the wallet.
- **keystore.jks** and **truststore.jks**: Java keystore and truststore files. They are protected by the wallet password provided while downloading the wallet.
- **ojdbc.properties**: Contains the wallet related connection property required for JDBC connection. This should be in the same path as **tnsnames.ora**.
- **README**: Contains wallet expiration information and links for Autonomous Database tools and resources.

See **Wallet README File** for information on the contents of the **README** file.

Notes for wallet files and the wallet password:

- Wallet files, along with the Database user ID and password provide access to data in your database. Store wallet files in a secure location. Share wallet files only with authorized users. If wallet files are transmitted in a way that might be accessed by unauthorized users (for example, over public email), transmit the wallet password separately and securely.

- For better security, Oracle recommends using restricted permissions on wallet files. This means setting the file permissions to 600 on Linux/Unix. Similar restrictions can be
achieved on Windows by letting the file owner have Read and Write permissions while all other users have no permissions.

- Autonomous Database uses strong password complexity rules for all users based on Oracle Cloud security standards. For more information on the password complexity rules see Create Users on Autonomous Database.

- The README file that contains wallet expiration information is not available in wallet zip files that were downloaded before April 2020.

- Starting six weeks before the wallet expiration date Autonomous Database sends notification emails each week, indicating the wallet expiration date. These emails provide notice before your wallet expires that you need to download a new wallet. You will receive these notification emails only if there is a connection that uses a wallet that is about to expire.

  You can also use the WalletExpirationWarning event to be notified when a wallet is due to expire. You will receive these notification events only if you are subscribed to Critical events and there is a connection that uses a wallet that is about to expire. See About Events Based Notification and Automation on Autonomous Database for more information.

### Connect Oracle SQL Developer with a Wallet (mTLS)

Oracle SQL Developer is a free integrated development environment that simplifies the development and management of Autonomous Database.

SQL Developer can connect to Autonomous Database and contains enhancements for key Autonomous Database features. You can download the latest version of Oracle SQL Developer for your platform from the Download link on this page: Oracle SQL Developer.

For connecting with mTLS authentication, Oracle SQL Developer provides support for wallet files using the Cloud Wallet Connection Type. Oracle recommends that you use version 18.2 (or later); however, earlier versions of SQL Developer will work with Autonomous Database using an Oracle Wallet.

For connecting with TLS authentication, Oracle SQL Developer provides support using the Custom JDBC Connection Type. See Connect with Oracle SQL Developer with TLS Authentication for details on connecting using TLS authentication.

To create a new mTLS connection to Autonomous Data Warehouse, do the following:

1. Start Oracle SQL Developer and in the connections panel, right-click Connections and select New Database Connection....
2. Choose the Connection Type **Cloud Wallet**.

3. Enter the following information:
   - **Connection Name**: Enter the name for this connection.
   - **Username**: Enter the database username. You can either use the default administrator database account (**ADMIN**) provided as part of the service or create a new schema, and use it.
   - **Password**: Enter the password for the database user.
   - **Connection Type**: Select **Cloud Wallet** (if you are using SQL Developer 18.2, this is Cloud PDB)
   - **Configuration File**: Click **Browse**, and select the client credentials zip file.
   - **Service**: Enter the database TNS name. The client credentials file includes a **tnsnames.ora** file that provides database TNS names with corresponding services.
4. Click **Connect** to connect to the database.

**Predefined Database Service Names for Autonomous Data Warehouse**

The tnsnames.ora file provided with the credentials zip file contains three database service names identifiable as **high**, **medium**, and **low**. The predefined service names provide different levels of performance and concurrency for Autonomous Data Warehouse.

- **high**: The High database service provides the highest level of resources to each SQL statement resulting in the highest performance, but supports the fewest number of concurrent SQL statements. Any SQL statement in this service can use all the CPU and IO resources in your database. The number of concurrent SQL statements that can be run in this service is 3, this number is independent of the number of OCPUs in your database.

- **medium**: The Medium database service provides a lower level of resources to each SQL statement potentially resulting a lower level of performance, but supports more concurrent SQL statements. Any SQL statement in this service can use multiple CPU and IO resources in your database. The number of concurrent SQL statements that can be run in this service depends on the number of OCPUs in your database.

- **low**: The Low database service provides the least level of resources to each SQL statement, but supports the most number of concurrent SQL statements. Any SQL statement in this service can use a single CPU and multiple IO resources in your database. The number of concurrent SQL statements that can be run in this service can be up to 300 times the number of OCPUs.

Sessions in these services may get disconnected if they stay idle for more than five (5) minutes and other users' sessions require the resources consumed by the idle session. This allows resources to be freed for other active users in your database.
The following shows the details for the number of concurrent statements for each connection service.

The number of OCPUs is the **CPU Core Count** shown in the Oracle Cloud Infrastructure Console.

The following table shows sample values for a database with 16 OCPUs.

<table>
<thead>
<tr>
<th>Database Service Name</th>
<th>Number of Concurrent Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>3</td>
</tr>
<tr>
<td>medium</td>
<td>20</td>
</tr>
<tr>
<td>low</td>
<td>Up to 300 times the number of OCPUs</td>
</tr>
</tbody>
</table>

Choose whichever database service offers the best balance of performance and concurrency.

**Note:**

When connecting for replication purposes, use the `low` database service name. For example use this service with Oracle GoldenGate connections.

See Manage Concurrency and Priorities on Autonomous Database for more information.
Load Data

Describes packages and tools to load data with Autonomous Data Warehouse.

Topics

• About Data Loading
• Load Data with Oracle Database Actions
• Load Data from Local Files with Oracle Database Actions
• Load Data from Files in the Cloud

About Data Loading

Autonomous Data Warehouse provides the following loading options:

• You can load data using Oracle Database Actions.
• You can load data using Oracle Database tools and Oracle or other 3rd party data integration tools.
• On transaction processing systems you traditionally ingest data through routine transactions or with DML operations.

In general you load data from files local to your client computer or from files stored in a cloud-based object store. To load data from files in the cloud, use either Oracle Database Actions or use the Autonomous Data Warehouse PL/SQL package `DBMS_CLOUD` to load files from the cloud.

For the fastest data loading experience Oracle recommends uploading the source files to a cloud-based object store, such as Oracle Cloud Infrastructure Object Storage, before loading the data into your Autonomous Data Warehouse database. Oracle provides support for loading files that are located locally in your data center, but when using this method of data loading you should factor in the transmission speeds across the Internet which may be significantly slower.

For more information on Oracle Cloud Infrastructure Object Storage, see Putting Data into Object Storage and Overview of Object Storage.

Load Data with Oracle Database Actions

Oracle Database Actions provides a web-based interface with development tools, data tools, administration, and monitoring features and lets you load or access data from local files, from cloud storage, or from remote databases.

On the Database Actions Data Load page, you can choose to load data from a file on your local device, from cloud storage, or from a database. You can also choose to explore the data in your Oracle Autonomous Database. See The Data Load Page for detailed information and the steps for loading data using Database Actions.
See Connect with Built-in Oracle Database Actions for information on accessing Oracle Database Actions.

Load Data from Local Files with Oracle Database Actions

Using Oracle Database Actions, from the Worksheet page, you can load data from local files into an existing table.

Topics

- Load Data into Existing Autonomous Database Table with Oracle Database Actions
Load Data into Existing Autonomous Database Table with Oracle Database Actions

You can load data into an existing table in Autonomous Database with the Database Actions import from file feature.

Before you load data, create the table in Autonomous Database. The file formats that you can upload with the Database Actions upload feature are CSV, XLS, XLSX, TSV and TXT.

To upload data from local files to an existing table with Database Actions, do the following:

1. Access Database Actions from the Oracle Cloud Infrastructure Console or with the Database Actions link provided to you.
2. To import data, in Database Actions, under Development click **SQL**. This shows an SQL worksheet.

3. In the **Navigator**, right-click the table where you want to load data.
4. In the menu select **Data loading** → **Upload Data**...
   
   For example, select the **SALES** table, right-click, and select **Data loading** → **Upload Data**...
5. In the Import data dialog you can either drag and drop files or click **Select files** to show a browser to select the files to import.

6. Complete the mapping for the columns you are importing. There are a number of options for column mapping. Click **(Show/Hide options)** icon to show the data import and format options to change column names, skip rows, rows to load, and various other options.

   Click **Apply** to apply the options you select.

7. When you finish selecting format and mapping options, click **Next** to preview the column mapping.
If there is a problem at this stage, information shows with more details, such as: **2 pending actions**. This means you need to correct or fix the source file data before you import.

8. Click **Next**.

9. Click **Next** to review the column mapping.

This shows the **Review** page to review the source columns and target columns for the import:

10. Click **Finish**.

11. Click **OK** to confirm the import.

   Depending on the size of the data file you are importing, the import may take some time.

Database Actions provides history to show the status of the import and to allow you to review the results or errors associated with the import operation.
For a detailed summary of the upload process, right-click the table in the Navigator tab, select Data loading, and then select Loaded Data. A summary of the data loaded is displayed in the Loaded data summary dialog.

If any data failed to load, you can view the number of rows in the Failed Rows column. Click the column and a dialog is displayed showing the failed rows.

In the Loaded data summary dialog, you can also search for files loaded by schema name, table name, or file name. To remove the loaded files, click the Delete icon.

Load Data from Files in the Cloud

The PL/SQL package `DBMS_CLOUD` provides support for loading data from text, ORC, Parquet, and Avro files in the Cloud to your tables in Autonomous Data Warehouse. In addition, `DBMS_CLOUD` you can load Data Pump dump files in the Cloud to your tables in Autonomous Data Warehouse.

The package `DBMS_CLOUD` supports loading files from the following cloud services:

- Oracle Cloud Infrastructure Object Storage
- Azure Blob Storage
- Amazon S3
- Amazon S3-Compatible, including: Google Cloud Storage and Wasabi Hot Cloud Storage.

Topics

- Create Credentials and Copy Data into an Existing Table
- Monitor and Troubleshoot Loads

Create Credentials and Copy Data into an Existing Table

For data loading from files in the Cloud, you need to first store your object storage credentials in your Autonomous Data Warehouse and then use the procedure `DBMS_CLOUD.COPY_DATA` to load data.

The source file in this example, `channels.txt`, has the following data:

```
S,Direct Sales,Direct
T,Tele Sales,Direct
C,Catalog,Indirect
I,Internet,Indirect
P,Partners,Others
```

1. Store your object store credentials using the procedure `DBMS_CLOUD.CREATES_CREDENTIAL`. For example:

   ```sql
   SET DEFINE OFF
   BEGIN
   DBMS_CLOUD.CREATES_CREDENTIAL(
      credential_name => 'DEF_CRED_NAME',
      username => 'adwc_user@example.com',
      password => 'password'
   );
   ```
This operation stores the credentials in the database in an encrypted format. You can use any name for the credential name. Note that this step is required only once unless your object store credentials change. Once you store the credentials you can then use the same credential name for all data loads.

Creating a credential to access Oracle Cloud Infrastructure Object Store is not required if you enable resource principal credentials. See Use Resource Principal to Access Oracle Cloud Infrastructure Resources for more information.

Note:

Some tools like SQL*Plus and SQL Developer use the ampersand character (&) as a special character. If you have the ampersand character in your password use the SET DEFINE OFF command in those tools as shown in the example to disable the special character and get the credential created properly.

2. Load data into an existing table using the procedure DBMS_CLOUD.COPY_DATA. For example:

```sql
CREATE TABLE CHANNELS
    (channel_id CHAR(1),
    channel_desc VARCHAR2(20),
    channel_class VARCHAR2(20)
);
/

BEGIN
    DBMS_CLOUD.COPY_DATA(
        table_name =>'CHANNELS',
        credential_name =>'DEF_CRED_NAME',
        file_uri_list =>'https://objectstorage.us-phoenix-1.oraclecloud.com/n/
                        namespace-string/b/bucketname/o/channels.txt',
        format => json_object('delimiter' value ',')
    );
END;
/
```

The parameters are:

- `table_name`: is the target table’s name.
- `credential_name`: is the name of the credential created in the previous step.
- `file_uri_list`: is a comma delimited list of the source files you want to load.
- `format`: defines the options you can specify to describe the format of the source file, including whether the file is of type text, ORC, Parquet, or Avro.

In this example, `namespace-string` is the Oracle Cloud Infrastructure object storage namespace and `bucketname` is the bucket name. See Understanding Object Storage Namespaces for more information.
Monitor and Troubleshoot Loads

All data load operations done using the PL/SQL package DBMS_CLOUD are logged in the tables dba_load_operations and user_load_operations:

- dba_load_operations: shows all load operations.
- user_load_operations: shows the load operations in your schema.

Query these tables to see information about ongoing and completed data loads. For example:

```sql
SELECT table_name, owner_name, type, status, start_time, update_time, logfile_table, badfile_table
FROM user_load_operations WHERE type = 'COPY';
```

Using this `SELECT` statement with a `WHERE` clause predicate on the `TYPE` column, shows load operations with the type `COPY`.

The `LOGFILE_TABLE` column shows the name of the table you can query to look at the log of a load operation. For example, the following query shows the log of the load operation:

```sql
select * from COPY$21_LOG;
```

The column `BADFILE_TABLE` shows the name of the table you can query to look at the rows that got errors during loading. For example, the following query shows the rejected records for the load operation:

```sql
select * from COPY$21_BAD;
```

Depending on the errors shown in the log and the rows shown in the specified `BADFILE_TABLE` table you can correct the error by specifying the correct format options in DBMS_CLOUD.COPY_DATA.

When the format type is "datapump", any rows rejected up to the specified `rejectlimit` are logged in the log file, but `badfiles` are not generated.

---

**Note:**

The `LOGFILE_TABLE` and `BADFILE_TABLE` tables are stored for two days for each load operation and then removed automatically.
Monitor and Troubleshoot ORC, Parquet, or Avro File Loading

As with other data files, ORC, Parquet, and Avro data loads generate logs that are viewable in the tables `dba_load_operations` and `user_load_operations`. Each load operation adds a record to `dba[<user>].load_operations` that indicates the table containing the logs.

The log table provides summary information about the load.

**Note:**

For ORC, Parquet, or Avro files, when the `format` parameter `type` is set to the value `orc`, `parquet` or `avro` the `BADFILE_TABLE` table is always empty.

- PRIMARY KEY constraint errors throw an ORA error.
- If data for a column encounters a conversion error, for example, the target column is not large enough to hold the converted value, the value for the column is set to `NULL`. This does not produce a rejected record.
Querying External Data with Autonomous Database

Describes packages and tools to query and validate data with Autonomous Data Warehouse.

Topics

- Query External Data
- Validate External Data
- View Logs for Data Validation

Query External Data

To query data in files in the Cloud, you need to first store your object storage credentials in your Autonomous Data Warehouse, and then create an external table using the PL/SQL procedure `DBMS_CLOUD.CREATE_EXTERNAL_TABLE`.

The procedure `DBMS_CLOUD.CREATE_EXTERNAL_TABLE` supports external files in the supported cloud object storage services, including:

- Oracle Cloud Infrastructure Object Storage
- Azure Blob Storage
- Amazon S3
- Amazon S3-Compatible, including: Oracle Cloud Infrastructure Object Storage, Google Cloud Storage, and Wasabi Hot Cloud Storage.

The source file in this example, `channels.txt`, has the following data:

S,Direct Sales,Direct
T,Tele Sales,Direct
C,Catalog,Indirect
I,Internet,Indirect
P,Partners,Others

1. Store your object store credentials using the procedure `DBMS_CLOUD.CREATE_CREDENTIAL`.

For example:

```sql
BEGIN
    DBMS_CLOUD.CREATE_CREDENTIAL(
        credential_name => 'DEF_CRED_NAME',
        username => 'adwc_user@example.com',
        password => 'password' );
END;
/
```
Creating a credential to access Oracle Cloud Infrastructure Object Store is not required if you enable resource principal credentials. See Use Resource Principal to Access Oracle Cloud Infrastructure Resources for more information.

This operation stores the credentials in the database in an encrypted format. You can use any name for the credential name. Note that this step is required only once unless your object store credentials change. Once you store the credentials you can then use the same credential name for creating external tables.

2. Create an external table on top of your source files using the procedure `DBMS_CLOUD.CREATE_EXTERNAL_TABLE`.

The procedure `DBMS_CLOUD.CREATE_EXTERNAL_TABLE` supports external files in the supported cloud object storage services. The credential is a table level property; therefore, the external files must be on the same object store.

For example:

```
BEGIN
    DBMS_CLOUD.CREATE_EXTERNAL_TABLE(
        table_name =>'CHANNELS_EXT',
        credential_name =>'DEF_CRED_NAME',
        file_uri_list =>'https://objectstorage.us-phoenix-1.oraclecloud.com/n/namespace-string/b.bucketname/o/channels.txt',
        format => json_object('delimiter' value ','),
        column_list => 'CHANNEL_ID NUMBER, CHANNEL_DESC VARCHAR2(20), CHANNEL_CLASS VARCHAR2(20)' );
END;
/
```

The parameters are:

- `table_name`: is the external table name.
- `credential_name`: is the name of the credential created in the previous step.
- `file_uri_list`: is a comma delimited list of the source files you want to query.
- `format`: defines the options you can specify to describe the format of the source file.
- `column_list`: is a comma delimited list of the column definitions in the source files.

In this example, `namespace-string` is the Oracle Cloud Infrastructure object storage namespace and `bucketname` is the bucket name. See Understanding Object Storage Namespaces for more information.

You can now run queries on the external table you created in the previous step. For example:

```
SELECT count(*) FROM channels_ext;
```

By default the database expects all rows in the external data file to be valid and match both the target data type definitions as well as the format definition of the files. If there are any rows in the source files that do not match the format options you specified, the query reports an error. You can use `DBMS_CLOUD` parameters, like `rejectlimit`, to suppress these errors. As an alternative, you can also validate the
external table you created to see the error messages and the rejected rows so that you can change your format options accordingly. See Validate External Data for more information.

Validate External Data

To validate any external table, you can use the procedure DBMS_CLOUD.VALIDATE_EXTERNAL_TABLE.

Before validating an external table you need to create the external table. To create an external table use the procedure for your table type, either DBMS_CLOUD.CREATE_EXTERNAL_TABLE. For example:

BEGIN
    DBMS_CLOUD.VALIDATE_EXTERNAL_TABLE (
        table_name => 'CHANNELS_EXT'
    );
END;
/

This procedure scans your source files and validates them using the format options specified when you create the external table.

The validate operation, by default, scans all the rows in your source files and stops when a row is rejected. If you want to validate only a subset of the rows, use the rowcount parameter. When the rowcount parameter is set the validate operation scans rows and stops either when a row is rejected or when the specified number of rows are validated without errors.

For example, the following validate operation scans 100 rows and stops when a row is rejected or when 100 rows are validated without errors:

BEGIN
    DBMS_CLOUD.VALIDATE_EXTERNAL_TABLE (
        table_name => 'CHANNELS_EXT',
        rowcount => 100
    );
END;
/

If you do not want the validate to stop when a row is rejected and you want to see all rejected rows, set the stop_on_error parameter to FALSE. In this case VALIDATE_EXTERNAL_TABLE scans all rows and reports all rejected rows.

If you want to validate only a subset of rows use the rowcount parameter. When rowcount is set and stop_on_error is set to FALSE, the validate operation scans rows and stops either when the specified number of rows are rejected or when the specified number of rows are validated without errors. For example, the following example scans 100 rows and stops when 100 rows are rejected or when 100 rows are validated without errors:

BEGIN
    DBMS_CLOUD.VALIDATE_EXTERNAL_TABLE (
        table_name => 'CHANNELS_EXT',
        rowcount => 100,
        stop_on_error => FALSE
    );
See View Logs for Data Validation to see the results of validate operations in the tables `dba_load_operations` and `user_load_operations`.

View Logs for Data Validation

To validate an external table, use the procedures `DBMS_CLOUD.VALIDATE_EXTERNAL_TABLE`, `DBMS_CLOUD.VALIDATE_EXTERNAL_PART_TABLE`, and `DBMS_CLOUD.VALIDATE_HYBRID_PART_TABLE`.

After you validate your source files you can see the result of the validate operation by querying a load operations table:

- `dba_load_operations`: shows all validate operations.
- `user_load_operations`: shows the validate operations in your schema.

You can use these files to view load validation information. For example use this select operation to query `user_load_operations`:

```
SELECT table_name, owner_name, type, status, start_time, update_time, logfile_table, badfile_table
FROM user_load_operations
WHERE type = 'VALIDATE';
```

<table>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>OWNER_NAME</th>
<th>TYPE</th>
<th>STATUS</th>
<th>START_TIME</th>
<th>UPDATE_TIME</th>
<th>LOGFILE_TABLE</th>
<th>BADFILE_TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANNELS_EXT</td>
<td>SH</td>
<td>VALIDATE</td>
<td>COMPLETED</td>
<td>13-NOV-17...</td>
<td>13-NOV-17...</td>
<td>$21_LOG</td>
<td>$21_BAD</td>
</tr>
</tbody>
</table>

Using this SQL statement with the `WHERE` clause on the `TYPE` column displays all of the load operations with type `VALIDATE`.

The `LOGFILE_TABLE` column shows the name of the table you can query to look at the log of a validate operation. For example, the following query shows the log for this validate operation:

```
SELECT * FROM VALIDATE$21_LOG;
```

The column `BADFILE_TABLE` shows the name of the table you can query to look at the rows where there were errors during validation. For example, the following query shows the rejected records for the above validate operation:

```
SELECT * FROM VALIDATE$21_BAD;
```

Depending on the errors shown in the log and the rows shown in the `BADFILE_TABLE`, you can correct the error by dropping the external table using the `DROP TABLE` command and recreating it by specifying the correct format options in
DBMS_CLOUD.CREATE_EXTERNAL_TABLE, DBMS_CLOUD.CREATE_EXTERNAL_PART_TABLE or DBMS_CLOUD.CREATE_HYBRID_PART_TABLE.

Note:

The LOGFILE_TABLE and BADFILE_TABLE tables are stored for two days for each validate operation and then removed automatically.
Working with Analytics and Visualization

Gain insight into your data with Oracle Analytics Cloud and Oracle Analytics Desktop. These tools let you explore your Autonomous Data Warehouse data through advanced analytics and interactive visualizations.

Topics
- Using Oracle Analytics Desktop with Autonomous Data Warehouse
- Use Oracle Analytics Cloud with Autonomous Data Warehouse

Using Oracle Analytics Desktop with Autonomous Data Warehouse

You can use Oracle Analytics Desktop with Autonomous Data Warehouse. Built on a high-performance platform with flexible data storage, Oracle Analytics Desktop provides a complete set of tools for deriving and sharing data insights.

- Data preparation: Analysts can ingest, profile, and cleanse data using a variety of algorithms.
- Data flow: Analysts can prepare, transform and aggregate data, and then run machine-learning models at scale.
- Data discovery: Subject matter experts can easily collaborate with other business users, blending intelligent analysis at scale, machine learning, and statistical modeling.
- Data visualization: Analysts can visualize any data, on any device, on premises and in the cloud.
- Data collaboration: Large organizations and small teams can share data more simply, as you don't need to manage or consolidate multiple versions of spreadsheets.
- Data-driven: Application developers can utilize interfaces that enable them to extend, customize, and embed rich analytic experiences in the application flow.

Connect with Oracle Analytics Desktop

Oracle Analytics Desktop makes it easy to visualize your data so you can focus on exploring interesting data patterns. Just connect to Autonomous Data Warehouse, select the elements that you’re interested in, and let Oracle Analytics Desktop find the best way to visualize it. Choose from a variety of visualizations to look at data in a specific way.

For details on connecting Autonomous Data Warehouse with Oracle Analytics Desktop, see User’s Guide for Oracle Analytics Desktop.
Use Oracle Analytics Cloud with Autonomous Data Warehouse

You can use Oracle Analytics Cloud with Autonomous Data Warehouse. Oracle Analytics Cloud provides a complete set of tools for deriving and sharing data insights.

- **Data preparation**: Analysts can ingest, profile, and cleanse data using a variety of algorithms.
- **Data flow**: Analysts can prepare, transform and aggregate data, and then run machine-learning models at scale.
- **Data discovery**: Subject matter experts can easily collaborate with other business users, blending intelligent analysis at scale, machine learning, and statistical modeling.
- **Data visualization**: Analysts can visualize any data, on any device, on-premise and in the cloud.
- **Data collaboration**: Large organizations and small teams can share data more simply, as you don't need to manage or consolidate multiple versions of spreadsheets.
- **Data-driven**: Application developers can utilize interfaces that enable them to extend, customize, and embed rich analytic experiences in the application flow.

See *Visualizing Data and Building Reports in Oracle Analytics Cloud* for details on connecting Autonomous Data Warehouse with Oracle Analytics Cloud.
Creating Web and Mobile Applications with Oracle Application Express

Describes how to create applications with Oracle Application Express on Autonomous Data Warehouse.

Topics
- About Oracle Application Express
- Access Oracle Application Express Administration Services
- Create Oracle Application Express Workspaces in Autonomous Data Warehouse
- Access Oracle Application Express App Builder
- Create Oracle Application Express Developer Accounts
- Use Web Services with Oracle Application Express
- Send Email from Oracle Application Express
- Restrictions and Limitations for Oracle Application Express with Autonomous Data Warehouse

About Oracle Application Express

Oracle Application Express (APEX) is a low-code development platform that enables you to build scalable, secure enterprise applications with world-class features that can be deployed anywhere.

Oracle APEX provides you with an easy-to-use browser-based environment to load data, manage database objects, develop REST interfaces, and build applications which look and run great on both desktop and mobile devices. You can use Oracle APEX to develop a wide variety of solutions: import spreadsheets and develop a single source of truth in minutes, create compelling data visualizations against your existing data, deploy productivity applications to elegantly solve a business need, or build your next mission-critical data management application.

Oracle APEX embraces SQL. Anything you can express with SQL can be easily employed in an Oracle APEX application. Oracle APEX also embodies low code with powerful data management and data visualization components, as well as responsive development out of the box. Instead of writing code by hand, you are able to use intelligent wizards to guide you through the rapid creation of applications and components.

Oracle APEX on Autonomous Database provides a preconfigured, fully managed and secured environment to both build and deploy world-class data-centric applications. There are no limits on the number of developers or end users for your Oracle APEX applications; Autonomous Database can instantly scale compute and storage online as needed, based upon your workload. Additionally, Oracle APEX applications developed on-premise can be easily deployed to Oracle APEX on Autonomous Database, or vice-versa.
Configuration, patching, monitoring, and upgrading of all Oracle Application Express components is fully managed by Oracle, leaving you free to focus on developing your solutions and solving your business problems. With Oracle APEX and low code, your organization can be more agile and develop solutions faster, for less cost, and with greater consistency. You can adapt to changing requirements with ease. And you can empower professional developers and everyone else in your organization to be a part of the solution.

This chapter covers information on Oracle Application Express specific to working on Autonomous Data Warehouse.

For more information on APEX, see the following:

- Oracle Application Express Release 21.1
- apex.oracle.com

Access Oracle Application Express Administration Services

Each Autonomous Data Warehouse instance includes a dedicated instance of Oracle Application Express; you can use this instance to create multiple workspaces. A workspace is a shared work area where you can build applications. You create workspaces in Application Express Administration Services.

Note:

If your Autonomous Data Warehouse is configured to use a Private Endpoint, then you can only access Oracle Application Express from clients in the same Virtual Cloud Network (VCN).

Perform the following prerequisite steps as necessary:

- Open the Oracle Cloud Infrastructure Console by clicking the next to Oracle Cloud.
- From the Oracle Cloud Infrastructure left navigation menu click Oracle Database, and then click Autonomous Data Warehouse.
- On the Autonomous Databases page select an Autonomous Database from the links under the Display Name column.

1. To access Application Express Administration Services you can use the Oracle Cloud Infrastructure console or the Autonomous Data Warehouse Service Console.

   To access Application Express Administration Services from Autonomous Data Warehouse Service Console:
   a. On the Autonomous Database details page click Service Console.
   b. Click Development.
   c. Click APEX.

   To access Application Express Administration Services from the Oracle Cloud Infrastructure console:
   a. On the Autonomous Database details page click the Tools tab.
b. In the Oracle Application Express area, click **Open APEX**.

The Application Express Administration Services sign-in page appears.

If you already created a workspace, the Application Express workspace sign-in page appears instead. To open Administration Services, click Administration Services link.

2. In the **Password** field, enter the password for the Autonomous Data Warehouse ADMIN user.

   **Note:**

   Application Express Administration Services and the Oracle Application Express development environment on Autonomous Database use Database Accounts authentication. This authentication method uses the database account user name and password to authenticate users.

3. Click **Sign In to Administration**.

   When you sign in for the first time, follow the prompts to create an Application Express workspace. See [Create Oracle Application Express Workspaces in Autonomous Data Warehouse](#) for more information.
You can also use Administration Services to manage your Application Express instance. See Oracle Application Express Administration Services in Oracle Application Express Administration Guide for more information.

Create Oracle Application Express Workspaces in Autonomous Data Warehouse

An Autonomous Data Warehouse instance does not have precreated Oracle Application Express workspaces. Create a workspace if you have not already done so or use these instructions to create additional workspaces.

To create an Oracle Application Express workspace:

1. Sign in to Application Express Administration Services.
   See Access Oracle Application Express Administration Services for more information.

2. Click Create Workspace.

3. On the Create Workspace page, in the Database User field, enter a new database username or choose an existing user from the list.
   The ADMIN database user cannot be associated with a workspace.

4. In the Password field, provide a strong password if the database user is a new user. If the user is an existing database user you do not enter a password.
5. (optional) In the **Workspace Name** field, change the name of the workspace that was automatically populated.

6. Click **Create Workspace**.

   See [Access Oracle Application Express App Builder](#) and [Create Oracle Application Express Developer Accounts](#) to create additional developer accounts.

---

### Access Oracle Application Express App Builder

Use App Builder to create and manage Oracle Application Express applications and application pages. The App Builder home page displays all installed applications in the current Oracle Application Express workspace.

If your Autonomous Data Warehouse is configured to use a Private Endpoint, you can only access Oracle Application Express App Builder from clients in the same Virtual Cloud Network (VCN).

To access Oracle Application Express App Builder:

1. Sign in to Application Express using the workspace name, username, and password you specify when you create the workspace.

   **Note:**

   Oracle Application Express Administration Services and the Oracle Application Express development environment on Autonomous Database use Database Accounts authentication. This authentication method uses the database account user name and password to authenticate users.

2. On the Workspace home page, click the App Builder icon.

   See [Create Oracle Application Express Developer Accounts](#) to create developer accounts.

---

### Create Oracle Application Express Developer Accounts

Oracle Application Express developers need a developer account in each workspace where they wish to build applications. The initial developer account is created when you create a workspace (this account also has Workspace Administrator privilege). These steps show you how to create additional developer accounts for members of your team or reset their passwords. When you create a developer account, a corresponding database user is automatically created.

To create developer accounts and provide direct access to Application Express:

1. Sign in to Application Express Administration Services.
2. Click **Manage Workspaces**.
3. Under Workspace Actions, click **Manage Developers and Users**.
4. On the Manage Application Developers and Users page, click **Create User**.
5. On the Create/Edit User page, in the **Username** field, enter a username.
6. In the **Email Address** field, enter an email address.
7. (Optional) Use the on-screen and in-line help to fill in additional fields.
8. In the User is an administrator field, select No.
9. In the User is a developer field, select Yes.
10. In the Password field, enter a strong password.
11. In the Confirm Password field, confirm the password.
12. At the top of the page, click Create User.

Alternatively, click Create and Create Another if you want to create the user and create another user.

To share sign-in details with developers:
1. On the Autonomous Databases page, under the Display Name column, select an Autonomous Database.
2. On the Autonomous Database Details page click Service Console.
3. Click Development.
4. Right-click APEX and choose Copy URL.
5. Provide the copied URL, along with the Workspace Name, the Username, and the Password for the developer account you created.

Using this URL developers can access the Application Express environment without having to navigate to the Autonomous Data Warehouse Service Console.

![Note:](#)

Changing the password of Workspace Administrators and Developers through Manage Developers and Users page or Edit Profile page only affects applications configured with "Application Express Accounts" authentication scheme. To change the password used to access App Builder, use Database Actions or another client to change the password of the corresponding database user.


Use Web Services with Oracle Application Express

You can interact with both SOAP and RESTful style web services from Application Express in your Autonomous Data Warehouse instance. Web services enable applications to interact with one another over the web in a platform-neutral, language independent environment. In a typical web services scenario, a business application sends a request to a service at a given URL by using the HTTP protocol. The service receives the request, processes it, and returns a response. Web services are typically based on Simple Object Access Protocol (SOAP) or Representational State Transfer (REST) architectures.

Using REST Data Sources (formerly called Web Source Modules), Application Express developers can declaratively access data services from a variety of REST endpoints, allowing both read and write operations. In addition to supporting smart caching rules for remote REST data, Oracle Application Express also offers the unique
ability to directly manipulate the results of REST data sources using industry standard SQL.

The APEX_WEB_SERVICE package enables you to integrate other systems with Application Express by allowing you to interact with web services anywhere you can use PL/SQL in your application. The package contains procedures and functions to call both SOAP and RESTful style web services, and to simplify implementation of OAuth 2.0 flows.

Note the following when working with web services in Application Express with Autonomous Data Warehouse:

• All web services must be secured. Only HTTPS services are supported on the default port (443). Your Application Express instance is pre-configured with an Oracle Wallet that contains more than 90 of the most common trusted root and intermediate SSL certificates. The APEX_WEB_SERVICE package automatically takes advantage of this Oracle Wallet without additional configuration from application developers.

• All web services must be accessible over the internet. The Autonomous Data Warehouse database is unable to reach web services deployed on private subnets or behind on-premises firewalls.

• Each Autonomous Data Warehouse instance is preconfigured with a network access control list (ACL) to permit outbound web service calls from Application Express. No further configuration by application developers is necessary.

• Your Application Express instance does not require an outbound web proxy.

• There is a default limit of 50,000 outbound web service requests per Application Express workspace in a rolling 24-hour period. If the limit of outbound web service calls is reached, the following SQL exception is raised on the subsequent request and the request is blocked:

  ORA-20001: You have exceeded the maximum number of web service requests per workspace. Please contact your administrator.

  You can raise the default limit up to 1,000,000 outbound web service requests by setting a value for the MAX_WEBSERVICE_REQUESTS parameter. For example, to change the limit to 250,000, connect to your Autonomous Data Warehouse database as ADMIN using a SQL client and execute the following:

  BEGIN
    APEX_INSTANCE_ADMIN.SET_PARAMETER('MAX_WEBSERVICE_REQUESTS', '250000');
  COMMIT;
  END;
 /

To learn more, see:

• APEX_WEB_SERVICE in Oracle Application Express API Reference

• Managing REST Data Sources in Oracle Application Express App Builder User's Guide
Send Email from Oracle Application Express

You can use the APEX_MAIL package to send emails from Oracle Application Express applications deployed in Autonomous Data Warehouse.

Before you use APEX_MAIL you must configure an email provider in your Application Express instance. Currently, the only supported email provider is Oracle Cloud Infrastructure Email Delivery service.

**Note:**
Third-party email providers are not supported.

To enable APEX_MAIL functionality in your Application Express instance in Autonomous Data Warehouse:

1. Identify the SMTP connection endpoint for Email Delivery. You configure the endpoint as the SMTP Host in your Application Express instance in Step 4. You may need to subscribe to additional Oracle Cloud Infrastructure regions if Email Delivery is not available in your current region. See Configure SMTP Connection for more information.

2. Generate SMTP credentials for Email Delivery. Your Application Express instance uses credentials to authenticate with Email Delivery servers when you send email. See Generate SMTP Credentials for a User for more information.

3. Create an approved sender for Email Delivery. You need to complete this step for all email addresses you use as the "From" with APEX_MAIL.SEND calls, as the Application Email From Address in your apps, or in the SMTP_FROM instance parameter. See Managing Approved Senders for more information.

4. Connect to your Autonomous Data Warehouse as ADMIN user using a SQL client and configure the following SMTP parameters using APEX_INSTANCE_ADMIN.SET_PARAMETER:

   - **SMTP_HOST_ADDRESS**: Specifies the SMTP connection endpoint from Step 1.
   - **SMTP_USERNAME**: Specifies the SMTP credential user name from Step 2.
   - **SMTP_PASSWORD**: Specifies the SMTP credential password from Step 2.
   - **Keep default values** for SMTP_HOST_PORT parameter (587) and SMTP_TLS_MODE parameter (STARTTLS).

For example:

```sql
BEGIN
    APEX_INSTANCE_ADMIN.SET_PARAMETER('SMTP_HOST_ADDRESS', 'smtp.us-phoenix-1.oraclecloud.com');
    APEX_INSTANCE_ADMIN.SET_PARAMETER('SMTP_USERNAME', 'ocid1.user.oc1.username');
    APEX_INSTANCE_ADMIN.SET_PARAMETER('SMTP_PASSWORD', 'password');
    COMMIT;
END;
/```
5. Validate the email configuration settings using a SQL client.

BEGIN
    APEX_INSTANCE_ADMIN.VALIDATE_EMAIL_CONFIG;
END;
/

If any errors are reported (for example, "ORA-29279: SMTP permanent error: 535 Authentication credentials invalid"), adjust the SMTP parameters and repeat the validation step.

6. Send a test email using APEX SQL Workshop, SQL Commands specifying one of the approved senders from Step 3 as "From". For example:

BEGIN
    APEX_MAIL.SEND(p_from => 'alice@example.com',
                    p_to   => 'bob@example.com',
                    p_subj => 'Email from Oracle Autonomous Database',
                    p_body => 'Sent using APEX_MAIL');
END;
/

7. To monitor email delivery in your Application Express instance:
   a. Sign in to APEX Administration Services.
   b. Open the Manage Instance page.
   c. Click the Mail Queue link in the Manage Meta Data section.

   Alternatively, query APEX_MAIL_QUEUE and APEX_MAIL_LOG views using a SQL client.

   **Note:**
   There is a limit of 5,000 emails per workspace in a 24-hour period. Oracle Cloud Infrastructure Email Delivery may impose additional limitations.

An approved sender must be set up for all "From:" addresses sending mail through Oracle Cloud Infrastructure, or mail will be rejected. There are limitations on approved senders with Oracle Cloud Infrastructure Email Delivery. See Managing Approved Senders for more information.

For more information, see:
- Overview of the Email Delivery Service
- APEX_MAIL in Oracle Application Express API Reference
- APEX_INSTANCE_ADMIN in Oracle Application Express API Reference
Restrictions and Limitations for Oracle Application Express with Autonomous Data Warehouse

This section lists the feature restrictions and limitations of Oracle Application Express when used within the context of Autonomous Data Warehouse. Certain limitations are required to protect the security and performance of your Oracle Application Express environment.

- **Disabled options in Administration Services:**
  Some Application Express Administration Services configuration options are disabled. The following are examples of configuration options that have been predefined by Oracle and cannot be altered:
  - Authentication scheme used to access App Builder ("Database Accounts").
  - Manage Requests - Ability to enable, submit, and approve self-service workspace requests and change requests. For example, Manage Instance, Self-Service Sign Up.
  - Manage Instance, Instance Settings - Security, Instance Settings, and Workspace Purge Setting. Workspace Isolation:
    * Workspace Purge Settings.
  - Web proxy, Oracle Wallet, and print server configuration
  - Daily limits of outbound web service calls and email messages
  - An option to make insecure outbound web service calls

- **Disabled options in Workspace Administration:**
  - Make a Service Request - Ability to make service requests for schemas, storage, and termination.

- **The following application authentication schemes are supported with limitations:**
  - HTTP Header Variable: Only after configuring a vanity URL. See Access Oracle Application Express, Oracle REST Data Services, and Developer Tools Using a Vanity URL to learn more.
  - LDAP Directory: including APEX_LDAP PL/SQL API, with the same restrictions that apply to DBMS_LDAP PL/SQL package. See Restrictions and Notes for Database PL/SQL Packages for more information.

- **Only the following APEX_INSTANCE_ADMIN procedures and functions are supported:**

<table>
<thead>
<tr>
<th>Procedure/Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD_SCHEMA</td>
</tr>
<tr>
<td>ADD_WORKSPACE</td>
</tr>
<tr>
<td>GET_PARAMETER</td>
</tr>
<tr>
<td>REMOVE_SCHEMA</td>
</tr>
</tbody>
</table>

---

Chapter 6
Restrictions and Limitations for Oracle Application Express with Autonomous Data Warehouse
6-10
See APEX_INSTANCE_ADMIN in Oracle Application Express API Reference for more information.

- SMTP Application Express instance parameters and certain others may be set using the APEX_INSTANCE_ADMIN package.
  See Available Parameter Values in Oracle Application Express API Reference for more information.
- Oracle Application Express is only available as a Full Development environment. Converting into a Runtime environment is not supported.