Oracle® Cloud

Using Oracle Database Exadata Express Cloud Service
# Contents

## Preface

<table>
<thead>
<tr>
<th>Audience</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related Resources</td>
<td>ix</td>
</tr>
<tr>
<td>Conventions</td>
<td>x</td>
</tr>
</tbody>
</table>

## 1 Get Started

<table>
<thead>
<tr>
<th>About Oracle Database Exadata Express Cloud Service</th>
<th>1-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture Overview</td>
<td>1-1</td>
</tr>
<tr>
<td>Key Features of Exadata Express</td>
<td>1-3</td>
</tr>
<tr>
<td>Before You Begin with Oracle Database Exadata Express Cloud Service</td>
<td>1-6</td>
</tr>
<tr>
<td>How to Begin with Oracle Database Exadata Express Cloud Service</td>
<td>1-7</td>
</tr>
<tr>
<td>Access My Services</td>
<td>1-7</td>
</tr>
<tr>
<td>Access Service Details</td>
<td>1-8</td>
</tr>
<tr>
<td>Access the Service Console</td>
<td>1-9</td>
</tr>
<tr>
<td>Access Oracle Application Express</td>
<td>1-11</td>
</tr>
<tr>
<td>Quick Tour of the Service Console</td>
<td>1-14</td>
</tr>
<tr>
<td>About the Service Console</td>
<td>1-14</td>
</tr>
<tr>
<td>About Welcome Tour</td>
<td>1-15</td>
</tr>
<tr>
<td>About Develop</td>
<td>1-17</td>
</tr>
<tr>
<td>About Application Express</td>
<td>1-20</td>
</tr>
<tr>
<td>About Client Development</td>
<td>1-20</td>
</tr>
<tr>
<td>About Manage</td>
<td>1-24</td>
</tr>
</tbody>
</table>

## 2 Manage Identity Domains and Service Instances

<table>
<thead>
<tr>
<th>About Exadata Express and Cloud Identity Domains</th>
<th>2-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Service Users, Roles and Privileges</td>
<td>2-2</td>
</tr>
<tr>
<td>Understand Service Details and Metrics</td>
<td>2-4</td>
</tr>
<tr>
<td>Create a Service Instance</td>
<td>2-5</td>
</tr>
<tr>
<td>Delete a Service Instance</td>
<td>2-7</td>
</tr>
<tr>
<td>Lock a Service Instance</td>
<td>2-8</td>
</tr>
</tbody>
</table>
# Connect Tools and Utilities

About Connecting Tools and Utilities 3-1  
Connect SQL*Plus 3-1  
  - Connect Using SQL*Plus from Oracle Instant Client 3-1  
  - Connect Using SQL*Plus from Oracle Database Client 3-3  
Connect SQLcl 3-4  
  - Download and Set up Oracle SQLcl 3-4  
  - Configure Oracle SQLcl Cloud Connection 3-5  
  - Invoke SQLcl from Oracle Developer Cloud Service 3-6  
Connect SQL Developer 3-6  
  - Install Oracle SQL Developer 3-6  
  - Configure Oracle SQL Developer Cloud Connection 3-7

# Connect Apps, Clients and Drivers

About Database Clients and Versions 4-1  
Database Client Prerequisites 4-1  
  - About Database Client Prerequisites 4-1  
  - Enable Oracle Net Services (SQL*Net) Access for Client Applications 4-2  
  - Download Client Credentials 4-3  
Connect with JDBC Thin Driver and UCP 4-4  
  - Software Requirements for JDBC Thin and UCP Connections 4-4  
  - Property Settings for JDBC Thin Driver and UCP 4-4  
Connect with Microsoft .NET and Visual Studio 4-7  
Connect with Node.js 4-7  
  - Connect an App with Node.js 4-7  
  - Connect with Node.js in ACCS 4-9

# Develop Document Store Applications Using SODA

About Using the Oracle Database as a Document Store through SODA 5-1  
Enable SODA for Your Service 5-2  
Use SODA for REST 5-5  
  - About SODA for REST 5-5  
  - SODA for REST Prerequisites 5-6  
Set up Users and Roles for SODA for REST 5-6  
  - Assign User Roles for SODA for REST 5-6  
  - Create Custom Roles 5-8  
About Authentication Mechanisms 5-11  
Determine the SODA for REST HTTP Operation URIs 5-11  
Get Started with SODA for REST Operations 5-12
6 Develop on the Database

- Develop with Oracle SQL Developer
  - Use Database Copy for Data Load 6-1
  - Use SQL Developer Cart for Data Load 6-3
  - Use Drag and Drop for Data Load 6-6
  - Execute SQL Scripts for Data Loading 6-8
- Use Oracle Application Express Application Data Load Utility 6-8
- Develop with Oracle SQLcl
  - Manage Database Objects with SQLcl 6-9
  - Query and Manipulate Data Using SQLcl 6-10
  - Use the help command in SQLcl 6-16
  - Use SQLcl to Data Load 6-18
- Develop with Oracle SQL*Plus
  - Enter and Execute Commands 6-19
  - Manage Database Objects with SQL*Plus 6-20
  - Query and Manipulate Data Using SQL*Plus 6-21
  - Use SQL*Plus to Data Load 6-23
- Manage Data with Oracle Application Express SQL Workshop
  - Use Oracle Application Express SQL Workshop to Develop the Database 6-24
  - Use Oracle Application Express SQL Workshop Data Upload Utility 6-25
  - Use Oracle Application Express Data Upload with Spreadsheet 6-25
  - Use Oracle Application Express SQL Workshop to Export Data 6-27
  - Upload Data with SQL Scripts 6-27

7 Manage Web Services

- Call SOAP and RESTful Web Services 7-1
- Implement RESTful Web Services with Oracle SQL Developer 7-2
  - Install and Administering REST Data Services 7-3
  - Auto REST Enabling Database Objects with Oracle SQL Developer 7-3
  - Develop RESTful Web Services with Oracle SQL Developer 7-4
- Implement RESTful Web Services with Oracle Application Express 7-5
- Consume RESTful Web Services with JET 7-5
8  Rapidly Developing Web Applications using Application Express

Manage Oracle Application Express Users 8-1
   About Oracle Application Express User Types 8-1
   Manage Oracle Application Express Users, Roles and Privileges 8-2
Manage Oracle Application Express Database Applications 8-3
   Create a Database Application 8-3
   Modify a Database Application 8-4
   Remove a Database Application 8-4
Manage Oracle Application Express Websheet Applications 8-5
   Create Websheet Database Objects 8-5
   Create a Websheet Application 8-6
   Modify a Websheet Application 8-6
   Remove a Websheet Application 8-6
Provide Access to Applications 8-7
Monitor Oracle Application Express Application Activity 8-7
Manage Oracle Application Express Packaged Apps and Sample Code 8-8
   Install Packaged Apps and Sample Code 8-9
   Lock and Unlock Packaged Apps 8-10
   Modify Packaged Apps and Sample Code 8-10
   Remove Packaged Apps and Sample Code 8-10
Manage Application Security 8-11
Use Oracle Application Express Applications and Websheets 8-11
Send Emails 8-11
Application Archiving 8-12
   Schedule Daily Automatic Application Archiving 8-13
   Manage and View Application Archives 8-13
      Install the Oracle APEX Application Archive Packaged App 8-14
      Run the Oracle APEX Application Archive App 8-15
      View and Manage Application Archives 8-16
   Manually Archiving Applications 8-16
   Restore Oracle APEX Applications 8-17

9  Manage and Monitor Database Performance

Use the Cloud Portal to Monitor Database Performance 9-1
Use SQL Developer to Monitor Database Performance 9-4
   Use Real Time SQL Monitor in Oracle SQL Developer 9-4
   View Database Status in Oracle SQL Developer 9-5
10 Manage Database Schemas

Find the Default Schema Name and Set the Password 10-1
Create Database Schemas 10-2
  Create a Database Schema Using the Service Console 10-3
  Create a Database Schema Using Oracle SQL Developer 10-4
  Create a Database Schema Using Oracle SQLcl 10-4
  Create a Database Schema Using Oracle SQL*Plus 10-5
Manage Database Schema Association with Oracle Application Express 10-6
  Create an Association between a Database Schema and Oracle Application Express 10-6
  Remove an Association between a Database Schema and Oracle Application Express 10-7
Export Database Schemas to Oracle Cloud Infrastructure Object Storage Classic 10-7
Import Database Schemas from Oracle Cloud Infrastructure Object Storage Classic 10-10
Post Import Tasks 10-13
  Validate Database Privileges and Object Dependencies 10-13
  Install APEX Applications 10-14
  Restore Oracle Application Express based RESTful Web Services 10-15
  Verify Oracle REST Data Services 10-16
Migrate Database Schema Service Schemas to Exadata Express 10-16

11 Migrate Applications and Data

About Migrating Applications 11-1
Migrate On-premises Applications to Oracle Database Exadata Express Cloud Service 11-1
Migrate Oracle Database Exadata Express Cloud Service Applications to On-premises Oracle Databases 11-3

12 Manage Vanity URLs for Exadata Express Hosted Applications

About Vanity URLs 12-1
Configure Vanity URLs 12-3
  Workflow for Configuring Vanity URLs 12-3
  Obtain the Public SSL Certificate from Exadata Express 12-4
  Configure Oracle Cloud Infrastructure Load Balancing Classic Instance 12-4
  Configure Vanity URLs from the Exadata Express Service Console 12-6
  Configure an IP Whitelist Policy 12-7
Frequently Asked Questions (FAQ)

How do I add more database storage to Exadata Express? 13-1
How do I track the amount of storage used? 13-2
What is the quickest way to go directly to the service console? 13-2
What PDB functionality is not available for this service? 13-4
Is HTTP traffic over the internet to Exadata Express encrypted? How are SSL/TLS certificates managed? 13-4
Can I run APEX applications on Exadata Express from our own domain name URL and SSL certificate? 13-5
How do I create an IP whitelist for my Exadata Express instance? 13-5
How do I stop billing in metered Exadata Express Cloud Service? 13-5
How do I manage new Exadata Express Cloud Service instance using Universal Credits? 13-5
Does the Exadata Express administrator need to perform Oracle Database backups? 13-6

Lockdown and Restrictions

Lockdown Profile A-1
Resource Restrictions A-2
Simultaneous Connection Considerations A-3
Preface

Using Oracle Database Exadata Express Cloud Service describes how to provision, monitor, and manage Oracle Database Exadata Express Cloud Service and provides references to documentation explaining how to develop and use Exadata Express applications.

Topics:
• Audience
• Related Resources
• Conventions

Audience

Using Oracle Database Exadata Express Cloud Service is intended for Oracle Database Exadata Express Cloud Service users who want to provision, monitor, and manage their services and develop or use Exadata Express applications.

Related Resources

For more information, see these Oracle resources:
• Oracle Public Cloud
  https://cloud.oracle.com
• Welcome to Oracle Cloud
• Managing and Monitoring Oracle Cloud
• What's New for Oracle Database Exadata Express Cloud Service
• Known Issues for Oracle Database Exadata Express Cloud Service
• Using Oracle Java Cloud Service - SaaS Extension
• Oracle Application Express Administration Guide
• Oracle Application Express SQL Workshop Guide
• Oracle Application Express End User Guide
• Oracle Application Express API Reference
• Oracle Application Express Application Builder User's Guide
• Oracle Application Express Release Notes
• Oracle SQL Developer User's Guide
• Oracle Database JSON Developer's Guide
Conventions

Text conventions used in this document are described in this section.

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>
Get Started

This section provides a brief overview of Oracle Database Exadata Express Cloud Service, its key concepts, and an overview of developing applications for the service.

Topics:
- About Oracle Database Exadata Express Cloud Service
- Key Features of Exadata Express
- Before You Begin with Oracle Database Exadata Express Cloud Service
- How to Begin with Oracle Database Exadata Express Cloud Service
- Access My Services
- Access Service Details
- Access Oracle Application Express
- Access the Service Console
- Access Oracle Application Express
- Quick Tour of the Service Console

To learn more about the Oracle Cloud, see Getting Started with Oracle Cloud.

About Oracle Database Exadata Express Cloud Service

Oracle Database Exadata Express Cloud Service delivers a managed Oracle Database 12c Release 2 Enterprise Edition with options, running on Oracle Exadata engineered systems – all at a highly affordable entry-level price. It is a full Oracle Database experience, compatible with what you get on-premises or in other Oracle Database Cloud Services, provisioned within minutes, and suitable for small to medium sized data.

For a description of Oracle Database 12c Release 2 features available in Exadata Express, see Key Features of Exadata Express.

For a description of new features available for each new release of Exadata Express, see What's New for Oracle Database Exadata Express Cloud Service.

For further information, go to the Exadata Express page on the Oracle Help Center for Cloud.

Architecture Overview

Each Exadata Express account provides one or several managed Pluggable Databases (PDBs) per account, of Oracle Database Enterprise Edition and relevant options, running on Exadata in Oracle Cloud. The number of PDBs depend on how many PDBs a customer purchases.

The service supports:
• Connecting database clients including language drivers, developer IDEs, and command-line tools
• Accessing and manipulating your data using REST interfaces
• Rapidly developing web applications in Oracle Application Express (APEX)

Many administrative tasks are taken care of for you in this managed service. Users can perform self-service administration through the web console, which provides access to the main functions of the service.

Database Clients

Oracle Database clients connect to Exadata Express over Oracle Net Services (SQL*Net). Oracle Connection Manager (CMAN) cluster running in Oracle Cloud arranges these connections. SQL*Net connections are encrypted and mutually authenticated using TLS 1.2. Exadata Express customers receive a public IP address, a SQL*Net port number, and a set of credentials to configure on the client. Connections can be established from the internet or other Oracle Cloud PaaS and IaaS services. Once connected, clients authenticate to the database as a standard user or a pre-defined administrator account (PDB_ADMIN).
REST Interfaces

Exadata Express users can configure REST endpoints using Oracle SQL Developer or APEX. REST connections go to Oracle REST Data Services (ORDS) running on an Oracle WebLogic Server (WLS) cluster in Oracle Cloud. This application tier is shared between customers and is managed by Oracle. From there, the connections route to a Container Database (CDB) and then on to the PDB of the user. Only HTTPS connections are supported.

ORDS can return responses in JSON format and other data format. Also, you optionally can configure Simple Oracle Document Access (SODA) to provide document-style interfaces over REST. REST endpoints can be protected using OAuth2 authentication.

APEX Applications

Each customer PDB includes a dedicated Oracle Application Express (APEX) instance, which is configured and managed by Oracle. APEX is a native low code framework of the Oracle Database for rapid development and deployment of modern, responsive, and scalable web applications. APEX applications are developed in a browser-based environment without the requirement of additional tools. Access to APEX is only enabled over HTTPS protocol.

Service Console

The service console is a web application for administering Exadata Express. The service console provides access to the main functions of the service. Users can set or reset the PDB_ADMIN password, enable or disable client access, quickly create a database schema, and so on.

Cloud Database

Exadata Express provides Oracle Database in an Oracle Multitenant architecture. Each user account consists of one or several dedicated PDB within a multitenant CDB. The CDBs run on bare metal Oracle Exadata machines in Oracle Cloud. Because Exadata Express is based on Oracle Database Enterprise Edition, users get advanced data management, analytics, and security capabilities (For example, data partitioning, data mining SQL, and data redaction, and so on). The largest Exadata Express shape includes Oracle Database In-Memory Column Store. Because the service runs on Exadata, it delivers unique features of Oracle engineered systems such as Hybrid Columnar Compression.

Key Features of Exadata Express

Some of the most important features of Oracle Database Exadata Express Cloud Service are described below. Note this list includes features of the service itself; database client tools and drivers that are updated for Exadata Express; and selected new features of Oracle Database 12c Release 1 and Release 2.

Application Development

- JSON Document Storage

  Oracle Database in Exadata Express provides direct storage, access and management of JSON documents. See JSON Support in Oracle Database New Features Guide 12c Release 2 (12.2).
• **Document-Style Data Access**

Oracle Database in Exadata Express gives you the ability to store and access data as schema-less documents and collections using the Simple Oracle Document Access (SODA) API. See Develop Document Store Applications Using SODA.

• **Oracle REST Data Services 3**

Exadata Express includes the newest Oracle REST Data Services (ORDS). With ORDS 3, it's easy to develop modern RESTful interfaces for relational data and now JSON documents stored in Oracle Database.

For relational data, ORDS 3 automatically generates REST endpoints for about a dozen single-table operations including create, query, update and delete. Easy-to-create JSON filters allow query predicates and sorts to be specified in a query-by-example format. For more complex operations, REST calls are mapped to SQL and PL/SQL routines you can write that will return data in JSON and other formats.

ORDS 3 also enables a wide range of operations to be performed directly on stored JSON documents in Oracle Database using the Simple Oracle Document Access (SODA) API over REST. These operations include create, query, update and delete with support for JSON filters. See Manage Web Services and Use SODA for REST. For more on ORDS 3, see Oracle REST Data Services on Oracle Technology Network.

• **Oracle Application Express 18.2**

Exadata Express includes Oracle Application Express (APEX) 18.2. APEX is a low-code environment to rapidly develop and deploy web apps for desktop and mobile browsers. You develop entirely within a browser, so no extra tools are required. See Rapidly Developing Web Applications using Application Express, for details on developing Application Express apps. See New Features in Oracle Application Express Release Notes to learn about major improvements introduced in this release.

• **SQL Developer 4 (Cloud Update)**

Oracle SQL Developer and SQL Developer Data Modeler provide a full environment for writing SQL and PL/SQL. These free IDEs not only aid in database development, they also can perform administrative tasks such as migrating data into Exadata Express from an on-premises database. They have been updated for Exadata Express to include drag-and-drop capabilities and more. See Oracle SQL Developer on Oracle Technology Network.

• **New SQLcl Command-Line**

SQLcl is a new and greatly improved command-line interface for Oracle Database. It has been updated for Exadata Express and is downloadable from the Service Console. Customers who are familiar with SQL*Plus should take a look at SQLcl and consider using Oracle’s powerful next generation command-line tool. See Oracle SQLcl on Oracle Technology Network.

• **Node.js Oracle Database Driver (General Availability)**

A new JavaScript language driver provides access to your Oracle Database from Node.js applications. This free and open source driver has been handed over to the development community with major contributions from Oracle. See Node.js Developer Center on Oracle Technology Network.

**Data Analytics**

• **Analytic Views**
Analytic views provide a business intelligence layer over a star schema, making it easy to extend the data set with hierarchies, levels, aggregate data, and calculated measures. Analytic views allow data warehouse and business intelligence application developers to extend the star schema with time series and other calculations, making data more valuable to business users and eliminating the need to define calculations within the application. This feature includes the new DDL statements CREATE ATTRIBUTE DIMENSION, CREATE HIERARCHY and CREATE ANALYTIC VIEW and their related ALTER and DROP statements, new calculated measure expression syntax, and new data dictionary views. See Overview of Analytic Views in Oracle Database New Features Guide 12c Release 2 (12.2).

• Data Mining SQL

Oracle Database in Exadata Express provides powerful data mining algorithms that enable data scientists and business analysts to discover insights, make predictions and leverage their Oracle data and investment. For example, with Oracle Data Mining SQL (ODM), you can build and apply predictive models inside the Oracle Database to foresee customer behavior, target your best customers, develop customer profiles, identify cross-selling opportunities and detect anomalies and potential fraud. Algorithms are implemented as SQL functions and leverage the strengths of Oracle Database. You can mine data tables and views as well as star schema data including transactional data, aggregations, unstructured data and more. The ODM functions honor all data and user privileges and security policies. See Oracle Data Mining on Oracle Technology Network.

Data Security

• Data Redaction

Data Redaction in Exadata Express provides selective, on-the-fly redaction of sensitive data in SQL query results prior to display by applications so that unauthorized users cannot view the sensitive data. Simple declarative policies apply different data transformations such as partial, random, and full redaction. Data Redaction can be applied conditionally, based on different factors that are tracked by the database or passed to the database by applications such as user identifiers, application identifiers, or client IP addresses. Data Redaction minimizes changes to applications by preserving the original data type and formatting when transformed data is returned.

Database Optimization

• In-Memory Column Store

Exadata Express provides a shape that includes Oracle Database In-Memory Column Store (IMC). This unique technology has a dual-format architecture that enables database tables to be simultaneously represented in memory using traditional row format and a new in-memory column format. The Oracle Database optimizer automatically routes analytic queries to the column format and transactional queries to the row format, transparently delivering best-of-both-worlds performance. Oracle Database maintains full transactional consistency between row and column storage. The new column format is a pure in-memory format and is not persistent on disk, so there are no additional storage costs or storage synchronization issues. Deploying IMC with virtually any existing Oracle Database compatible application is as easy as flipping a switch – no application changes are required. See Oracle Database In-Memory on Oracle Technology Network.
• **Data Partitioning**

Partitioning in Exadata Express enhances the performance, manageability, and availability of a wide variety of applications and helps reduce the total cost of ownership for storing larger amounts of data. Partitioning allows Oracle Database tables, indexes, and index-organized tables to be subdivided into smaller pieces, enabling these objects to be managed and accessed at a finer level of granularity. Oracle supports a rich variety of partitioning strategies and extensions to address diverse business requirements. Moreover, since it is entirely transparent, partitioning can be applied to almost any application data without the need for potentially expensive and time consuming application changes. See Oracle Partitioning on Oracle Technology Network.

• **Data Compression**

Data compression in Exadata Express provides a complete set of compression capabilities to help reduce storage consumption while maintaining fast performance. It allows customers to reduce their overall database storage footprint for relational and unstructured data. With Oracle Advanced Compression, you can reduce the storage required for transactional workloads by up to 2-4x. With Oracle Hybrid Columnar Compression, you can reduce the storage required for analytic workloads by up to 10x or more. Oracle’s unique data compression innovations not only reduce storage consumption, allowing you to store even more data in Exadata Express, but also keep the impact on query performance to a minimum. See Compression and Archiving in Oracle Database Concepts 12c Release 2 (12.2) and Advanced Index Compression in Oracle Database Administrator's Guide 12c Release 2 (12.2).

**News for Users of Existing Schema Service**

If you are a current customer of Oracle Database Cloud Service – Multitenant Edition (a.k.a. Database Schema Service), you may be considering migrating to Exadata Express. With Exadata Express, you can take advantage of all the features described above, plus you get:

- A dedicated Oracle Pluggable Database, with the ability to create multiple users and schemas - instead of being limited to a single schema.
- The ability to directly connect to your cloud database over Oracle Net Services (SQL*Net). This means you can leverage a large assortment of database client tools and programming language drivers - in addition to REST access and Oracle Application Express applications.

**Before You Begin with Oracle Database Exadata Express Cloud Service**

Developing applications for Oracle Database Exadata Express Cloud Service is done with Oracle Application Express or with IDEs that support standard database client drivers. Before building applications, you may want to familiarize yourself with the following procedures:

- Connecting applications, clients and drivers to your service. See About Database Clients and Versions.
- Working with JSON and other data using Oracle Document Store (SODA). See About Using the Oracle Database as a Document Store through SODA.
• Designing and populating the database. See Develop on the Database.

• Creating custom applications. See Manage Oracle Application Express Database Applications.

• Installing packaged applications. See Manage Oracle Application Express Packaged Apps and Sample Code.

• Creating RESTful Services. See Implement RESTful Web Services with Oracle SQL Developer.

• Managing application users. See Manage Oracle Application Express Users.

To learn more, see Rapidly Developing Web Applications using Application Express.

How to Begin with Oracle Database Exadata Express Cloud Service

Get started with the Oracle Database Exadata Express Cloud Service by signing up for a free credit promotion or purchase a subscription.

Here's how to get started with Exadata Express:

1. Learn about Universal Credits. See About Universal Credits in Getting Started with Oracle Cloud. Be sure to log in as a Traditional account user if you want to manage new Exadata Express instances. See How do I manage new Exadata Express Cloud Service instance using Universal Credits?

2. Sign up for a free credit promotion or purchase a subscription. See Requesting and Managing Free Oracle Cloud Promotions or Buying an Oracle Cloud Subscription in Getting Started with Oracle Cloud.

3. Activate and verify the service. See Activate Your Order or Activating Your Trial Subscription in Getting Started with Oracle Cloud.

4. If you are using trial subscriptions, verify activation. See Verifying That Your Trial Subscription Is Running in Getting Started with Oracle Cloud.

5. Learn about users and roles. See Manage Service Users, Roles and Privileges.

6. Create accounts for your users and assign them appropriate privileges and roles. See Oracle Cloud User Roles and Privileges in Getting Started with Oracle Cloud.

7. Set the password for the database user authorized to perform administrative tasks for your service (PDB_ADMIN). See Manage Service Users, Roles and Privileges.

Access My Services

The My Services page lists all of the service consoles you have access to.

To access My Services and drill down to the Service Details:

1. From your browser, go to the Oracle Cloud home page.
   The Oracle Cloud home page appears.

2. Click Sign In.
   The Oracle Cloud Sign In page displays.

3. Select the Data Center and click My Services.
4. Enter your Identity Domain and click Go.
5. Enter your username and password and click Sign In.
The My Services page displays.

Note:
When an Oracle Database Exadata Express Cloud Service is provisioned, a service instance is not automatically created. The first time you sign in to your My Services page you will not see a service instance displayed for your service. The service administrator must create the service instance. See Create a Service Instance.

Access Service Details

To navigate to the Service Details page for a particular Oracle Database Exadata Express Cloud Service, you start from My Services, locate the service you want to see details for and click on it.

To access the Service Details page for a particular Exadata Express service:

The My Services page displays.
2. Find the service you want to see Service Details for, and click the service link. The Service Details page displays.

Access the Service Console

From the Service Console, you can control web access, client-server access and you can perform service administration functions.

See About the Service Console for further details on functionality available from this page.
To access the Service Console:

1. Go to the Service Details for your service. See Access Service Details. The Service Details page displays.

2. Click Open Service Console. The Service Console appears.
Access Oracle Application Express

Your Oracle Database Exadata Express Cloud Service includes an Oracle Application Express instance. Accessing this rich development environment is easily done from the Service Console.

The Exadata Express service console provides options for Database and Web Application development using Oracle Application Express. The following Oracle Application Express components exposed from this page are grouped under Web Access:

• SQL Workshop
  The Oracle Application Express SQL Workshop Object Browser enables developers to browse, create, and edit objects in a database. To learn more, see Manage Data with Oracle Application Express SQL Workshop.

• Application Builder
  The Oracle Application Express Application Builder enables developers to manage Database and Websheet Applications. To learn more, see Manage Oracle Application Express Database Applications.

• RESTful Web Service
  The Oracle Application Express RESTful Web services enable the declarative specification of RESTful services used to access the database. These services work with Oracle REST Data Services to allow external applications to consume these services. To learn more, see Implement RESTful Web Services with Oracle Application Express.

• Packaged Applications
  An overview of the installed Packaged Apps and featured Packaged Apps are displayed. The Oracle Application Express Packaged Apps enables developers to view and install productivity applications from an application gallery. To learn more, see Manage Oracle Application Express Packaged Apps and Sample Code.

To access Oracle Application Express:

1. Go to the Service Details for the service where Oracle Application Express is located. See Access Service Details.

   The Service Details page displays.
2. Click **Open Service Console**.

   The Service Console appears.

3. Click Develop, on the Exadata Express Service Console.

   The four Oracle Application Express components display along with other categories.
a. Click **App Builder**.

**Tip:**
You can also access **App Builder** directly from Home page of Exadata Express's console by clicking on the arrow box next to APEX version.

The Oracle Application Express Application Builder home page appears.

b. Click **Productivity Apps**.

The Oracle Application Express Packaged Apps home page appears.

c. Click **SQL Workshop**.
The Oracle Application Express SQL Workshop home page displays.

d. Click **REST Data Services**.

The RESTful Services page of the Oracle Application Express SQL Workshop displays.

---

**Quick Tour of the Service Console**

The Oracle Database Exadata Express Cloud Service provides a Service Console, giving you a unified starting point to access and manage all aspects of your service.

**Topics**

- About the Service Console
- About Welcome Tour
- About Develop
- About Manage

**About the Service Console**

The options available on the Oracle Database Exadata Express Cloud Service Service Console are grouped into the following categories: Welcome Tour, Develop, and Manage. The service console also displays the database version, APEX version, and the storage information.

For instructions on how to access the Exadata Express Service Console, see Access the Service Console.
Note:

Only users with Database Administrator role in the Cloud Identity Management can see the Manage page.

About Welcome Tour

The Welcome Tour option from Oracle Database Exadata Express Cloud Service console provides an easy way to learn about the product. The Welcome Tour page provides short videos to learn about Exadata Express, setting up an administration password, enabling client access, creating new schema and loading data, and building Oracle Application Express (APEX) and Java applications.

Welcome Tour screen

The following image shows the video options under the Welcome Tour page.
<table>
<thead>
<tr>
<th>Option</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learn the Basics</td>
<td>Provides an overview of Exadata Express.</td>
</tr>
<tr>
<td>Set Password</td>
<td>Learn how to set administration password for your Exadata Express account. See <a href="#">Manage Service Users, Roles and Privileges</a>.</td>
</tr>
</tbody>
</table>
## Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect</td>
<td>Learn how to set up client access and enable client access. See <a href="#">Enable Oracle Net Services (SQL*Net) Access for Client Applications</a>.</td>
</tr>
<tr>
<td>Load Data</td>
<td>Learn how to create new schema and load your data. See <a href="#">Create Database Schemas</a>.</td>
</tr>
<tr>
<td>Develop</td>
<td>Learn how to build APEX applications and Java applications. See <a href="#">Rapidly Developing Web Applications using Application Express</a>.</td>
</tr>
</tbody>
</table>

### About Develop

The Develop page from Oracle Database Exadata Express Cloud Service console provides developer tools to create applications using the available programming languages or provides an easy access to develop using Oracle Application Express (APEX).

### Develop Page

The Develop page is divided in to the following sections: Application Express, Client Development, and Getting Started.
## Application Express Section

This section provides access to APEX applications.

### Develop

#### Application Express

<table>
<thead>
<tr>
<th>Option</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Builder</td>
<td>Develop and deploy applications from your browser. See About Application Express.</td>
</tr>
<tr>
<td>SQL Workshop</td>
<td>Learn and execute SQL commands. See About Application Express</td>
</tr>
<tr>
<td>Productivity Apps</td>
<td>Install from a various built-in productivity applications. See About Application Express</td>
</tr>
<tr>
<td>REST Data Services</td>
<td>Manage REST data interface. See About Application Express</td>
</tr>
</tbody>
</table>

#### Client Development Section

This section provides access to client development tools.
### Option Brief Description

#### Setup
Setup option in the Develop page provides information on the following:
- Client Access
- Client Credentials
See About Setup.

#### Tools and Utilities
Tools and Utilities option in the Develop page provides information on using the following tools and utilities for various tasks:
- SQL*Plus
- SQLcl
- SQL Developer
- Data Modeler
See About Tools and Utilities.

#### Languages and Framework
Languages and Framework option in the Develop page provides information on building applications in the following programming languages:
- Java
- .Net
- Node.js
- Oracle JET
- SQL and PL/SQL
- C and C++
- Python
- PHP
- REST
- Ruby
See About Languages and Framework.
Getting Started Section
The Getting Started section features some useful links directing the users to find relevant information.

About Application Express
The Application Express section from the Oracle Database Exadata Express Cloud Service console’s Develop page provides an easy way to access Oracle Application Express modules to develop. You can also access APEX App Builder directly from Home page of Exadata Express’s console by clicking on the arrow box next to APEX version.

Application Express options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Workshop</td>
<td>Allows you go directly to browser-based SQL Workshop, where you can run SQL statements, execute scripts and explore database objects. See Getting Started with SQL Workshop in Oracle Application Express SQL Workshop Guide.</td>
</tr>
<tr>
<td>App Builder</td>
<td>Quickly declaratively develop database and websheet applications. You can import files such as database applications and plug-ins. There is a dashboard showing metrics about your applications and workspace utilities to manage defaults, themes, metadata, exports, and more. See Rapidly Developing Web Applications using Application Express</td>
</tr>
<tr>
<td>REST Data Services</td>
<td>Directly access the page to define and manage RESTful web services that view and manipulate data objects within your database. See Manage Web Services</td>
</tr>
<tr>
<td>Productivity Apps</td>
<td>Install from a gallery of pre-built Oracle Application Express Productivity Apps. See Manage Oracle Application Express Packaged Apps and Sample Code</td>
</tr>
</tbody>
</table>

About Client Development
The Client Development section from the Oracle Database Exadata Express Cloud Service console’s Develop page allows you to easily set up client credentials and use...
other available tools and programming languages to build applications with Exadata Express.

**Client Development option**

<table>
<thead>
<tr>
<th>Option</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup</td>
<td>Setup option in the Develop page provides information on the following:</td>
</tr>
<tr>
<td></td>
<td>• Client Access</td>
</tr>
<tr>
<td></td>
<td>• Client Credentials</td>
</tr>
<tr>
<td></td>
<td>See About Setup.</td>
</tr>
<tr>
<td>Tools and Utilities</td>
<td>Tools and Utilities option in the Develop page provides information on using the</td>
</tr>
<tr>
<td></td>
<td>following tools and utilities for various tasks:</td>
</tr>
<tr>
<td></td>
<td>• SQL*Plus</td>
</tr>
<tr>
<td></td>
<td>• SQLcl</td>
</tr>
<tr>
<td></td>
<td>• SQL Developer</td>
</tr>
<tr>
<td></td>
<td>• Data Modeler</td>
</tr>
<tr>
<td></td>
<td>See About Tools and Utilities.</td>
</tr>
<tr>
<td>Languages and Framework</td>
<td>Languages and Framework option in the Develop page provides information on</td>
</tr>
<tr>
<td></td>
<td>building applications in the following programming languages:</td>
</tr>
<tr>
<td></td>
<td>• Java</td>
</tr>
<tr>
<td></td>
<td>• .Net</td>
</tr>
<tr>
<td></td>
<td>• Node.js</td>
</tr>
<tr>
<td></td>
<td>• Oracle JET</td>
</tr>
<tr>
<td></td>
<td>• SQL and PL/SQL</td>
</tr>
<tr>
<td></td>
<td>• C and C++</td>
</tr>
<tr>
<td></td>
<td>• Python</td>
</tr>
<tr>
<td></td>
<td>• PHP</td>
</tr>
<tr>
<td></td>
<td>• REST</td>
</tr>
<tr>
<td></td>
<td>• Ruby</td>
</tr>
<tr>
<td></td>
<td>See About Languages and Framework.</td>
</tr>
</tbody>
</table>
About Setup

The Setup option under the Client Development section allows you to set up client access and download client credentials.

Service Console Client Access options before client access has been enabled:

Service Console Client Access options available after enabling client access:

<table>
<thead>
<tr>
<th>Option</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Credentials</td>
<td>Download client credentials needed for clients to access your service. See Download Client Credentials.</td>
</tr>
<tr>
<td>Client Access</td>
<td>Use this option to disallow Oracle Net Services (SQL<em>Net) access to your service. This option is only available when client access has been enabled. See Enable Oracle Net Services (SQL</em>Net) Access for Client Applications.</td>
</tr>
</tbody>
</table>

About Tools and Utilities

The Tools and Utilities option under Client Development section allows you to access and download various SQL tools that can be used to connect to Exadata Express.

Tools and Utilities option:
### About Languages and Framework

The Languages and Framework option under the Client Development section allows you to build applications using various available programming languages.

#### Languages and Framework option

<table>
<thead>
<tr>
<th>Option</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>Provides access to Java Developer Tools that can be used with Exadata Express.</td>
</tr>
<tr>
<td>.Net</td>
<td>Provides access to .Net applications that can be used with Exadata Express.</td>
</tr>
<tr>
<td>Node.js</td>
<td>Provides access to OnPremises scripting languages that can be used with Exadata Express</td>
</tr>
<tr>
<td>Oracle JET</td>
<td>Provides access to Oracle JavaScript Extension Toolkit (JET) that can be used to build applications with Exadata Express</td>
</tr>
<tr>
<td>SQL and PL/SQL</td>
<td>Provides access to SQL and PL/SQL tools that can be used with Exadata Express.</td>
</tr>
<tr>
<td>C and C++</td>
<td>Provides access to Oracle C++ Call Interface (OCCI) that can be used with Exadata Express</td>
</tr>
<tr>
<td>Python</td>
<td>Provides access to OnPremises scripting languages that can be used with Exadata Express</td>
</tr>
<tr>
<td>Option</td>
<td>Brief Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------</td>
</tr>
<tr>
<td>PHP</td>
<td>Provides access to OnPremises scripting languages that can be used with Exadata Express.</td>
</tr>
<tr>
<td>REST</td>
<td>Provides access to Oracle REST Data Services that can be used with Exadata Express.</td>
</tr>
<tr>
<td>Ruby</td>
<td>Provides access to OnPremises scripting languages that can be used with Exadata Express.</td>
</tr>
</tbody>
</table>

**About Manage**

The Manage option from Oracle Database Exadata Express Cloud Service console provides access to perform management tasks for your cloud database and management tasks through Oracle Application Express (APEX). Only users with Database Administrator role in the Cloud Identity Management can see the Manage section.

**Manage option**

The Manage page is divided into the following sections: Database Management, Application Express, and Getting Started.
### Database Management Section

Provides access to perform management tasks on cloud database.

<table>
<thead>
<tr>
<th>Option</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin Password</td>
<td>Use this option to set the password for the PDB_ADMIN database user that is authorized to perform administrative tasks. See Manage Service Users, Roles and Privileges</td>
</tr>
<tr>
<td>Database Schema</td>
<td>Create a new schema for database objects. Note that a user and schema are related concepts. A user is the account you use to connect to the database schema with. Schema is the set of database objects, such as tables and views that belong to that user account. You can specify the schema name, schema password and choose whether to allow access from APEX, define a new tablespace, or enable REST data services. See Manage Database Schemas</td>
</tr>
<tr>
<td>Client Access</td>
<td></td>
</tr>
<tr>
<td>Client Credentials</td>
<td></td>
</tr>
<tr>
<td>Export to Cloud Storage</td>
<td></td>
</tr>
<tr>
<td>Import from Cloud Storage</td>
<td></td>
</tr>
<tr>
<td>Document Store</td>
<td>Create a document store for schema-less application development with JSON storage and access.</td>
</tr>
<tr>
<td>Option</td>
<td>Brief Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Document Store</td>
<td>This option enables you to create a document store, using either an existing schema or new schema, and to enable SODA for REST, which enables REST-based operations on the schema using Oracle’s SODA for REST API. It also enables SODA for Java, which is Oracle’s SODA for Java API for use with Java programs. See Enable SODA for Your Service</td>
</tr>
<tr>
<td>Export to Cloud Storage</td>
<td>Export database schemas to an Oracle Cloud Infrastructure Object Storage Classic container. See Export Database Schemas to Oracle Cloud Infrastructure Object Storage Classic.</td>
</tr>
<tr>
<td>Import from Cloud Storage</td>
<td>Import database schemas from Oracle Cloud Infrastructure Object Storage Classic container. See Import Database Schemas from Oracle Cloud Infrastructure Object Storage Classic.</td>
</tr>
</tbody>
</table>

**Application Express Section**

Provides access to perform management tasks through APEX.

<table>
<thead>
<tr>
<th>Option</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Administration</td>
<td>Manage schema association with APEX. See Manage Database Schema Association with Oracle Application Express.</td>
</tr>
<tr>
<td>App Archiving</td>
<td>Archive applications through APEX. See Application Archiving.</td>
</tr>
<tr>
<td>Vanity URL</td>
<td>Define a custom URL for your APEX instance and for your REST services. See Manage Vanity URLs for Exadata Express Hosted Applications.</td>
</tr>
</tbody>
</table>

**Getting Started Section**

The Getting Started section features some useful links directing the users to find relevant information.
Manage Identity Domains and Service Instances

An Oracle Database Exadata Express Cloud Service instance is an Oracle Database 12c Release 2 Pluggable Database (PDB). The service administrator is allowed to create, delete, and lock service instances and also has the ability to add users and monitor service metrics.

Topics

• About Exadata Express and Cloud Identity Domains
• Manage Service Users, Roles and Privileges
• Understand Service Details and Metrics
• Create a Service Instance
• Delete a Service Instance
• Lock a Service Instance

About Exadata Express and Cloud Identity Domains

If you have no other Oracle Cloud services, then each new Oracle Database Exadata Express Cloud Service you purchase will be provisioned into a new and distinct cloud identity domain (ID Domain). However, if you already have another Oracle Cloud service (platform or infrastructure), and you create an Exadata Express service for the first time, then Exadata Express will be provisioned automatically into your existing cloud ID Domain. This convenience makes Exadata Express more usable with other Oracle Cloud platform and infrastructure services.

Customers with multiple ID Domains may wish to consolidate their existing cloud services into fewer ID Domains. To move Exadata Express service from one ID Domain to another:

1. Sign in to Order Management / My Account as the Account Administrator.
2. Select the desired Exadata Express service.
3. Navigate to the Identity Domain Administration page.
4. Open the Relocate Service tab.
5. Select the target ID Domain.

During the relocation process, all access to the service will be disabled. The data will be retained and all access URLs (for example, to the Service Console) will not change.

One important restriction on this behavior is that you can have only one single Exadata Express service in each ID Domain. In other words, you can relocate one Exadata Express into an existing platform or infrastructure ID Domain (or benefit from automatic placement if it is a new Exadata Express account), but you cannot merge multiple Exadata Express services into a single ID Domain.
Manage Service Users, Roles and Privileges

All service users and roles defined as part of the Cloud Identity Domain are administered from the Security page in My Services. On this page, an Identity Domain or Service Administrator is allowed to add, delete and modify users, and to create, delete, or assign roles.

Identity Domain Administrators are allowed to access all users defined within their Identity Domain and their roles. Service Administrators only have access to the users defined for their service, and users of a service can only modify their own user profile and reset their account password. Oracle Database Exadata Express Cloud Service uses Traditional Cloud Accounts, as opposed to Cloud Accounts with Identity Cloud Service (IDCS).

For more details, refer to Adding Users and Assigning Roles in Getting Started with Oracle Cloud.

The following table lists user roles and privileges specific to Exadata Express Cloud Service.

<table>
<thead>
<tr>
<th>User Role</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database User</td>
<td>This role is used only when the customer builds an Oracle Application Express (APEX) application or REST service that requires end users to have this role. Users with this role cannot access service console or APEX App Builder.</td>
</tr>
<tr>
<td>Database Developer</td>
<td>Users with this role have access to service console and, when Client Access is enabled, can download client credentials and connect to the cloud database through SQL*Net. Users with this role become developers in Oracle Application Express (APEX).</td>
</tr>
</tbody>
</table>
| Database Administrator  | Manages administrative functions related to database using the service console. Users with this role become instance administrators in Oracle Application Express (APEX). A database administrator can:  
  • Reset a PDB_ADMIN password.  
  • Create a schema.  
  • Create a document store. |
| Entitlement Administrator| A user assigned to this role can create or delete cloud databases, based on your specific business requirements, but is restricted to a parent Oracle Cloud identity domain. |
### User Role

<table>
<thead>
<tr>
<th>Role</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity Domain Administrator</td>
<td>Performs user management functions that a service administrator performs, but restricted to the Oracle Cloud services within their identity domain. An identity domain administrator can:</td>
</tr>
<tr>
<td></td>
<td>• Create user accounts and roles for specific cloud databases.</td>
</tr>
<tr>
<td></td>
<td>• Create custom roles.</td>
</tr>
</tbody>
</table>

### Note:

When there are multiple Exadata Express cloud databases in a single Oracle Cloud identity domain then cloud users and roles must be assigned individually based on service instance name. These cloud users and roles are separate from database users and roles created within each cloud database.

### Managing the Administrator User (PDB_ADMIN)

The name of the privileged database user, authorized to perform administrative tasks for your service, is PDB_ADMIN. When your service instance is first created, the PDB_ADMIN user is locked and cannot be used to log in. To unlock this user, assign a new password by following the steps below.

**To change the password for PDB_ADMIN:**

1. Go to the Service Console. See Access the Service Console.
2. Under Administration options, click Set Administrator Password. The Set PDB_ADMIN Password dialog appears.
3. Make the following entries:
   - New password – enter a new password
   - Confirm password – re-enter the password to confirm
4. Click Set Password.

### Unlocking Your Account

The system automatically locks your user account if there are multiple incorrect sign-in attempts using your user name, password, and identity domain. If the PDB_ADMIN user becomes locked in the future (for example, after several invalid login attempts), change its password to unlock this user.

To unlock your account, follow the instructions at Resetting your password in Managing and Monitoring Oracle Cloud.
Understand Service Details and Metrics

The Oracle Database Exadata Express Cloud Service provides the ability for you to monitor important metrics for your service.

The Metrics available for the Oracle Database Exadata Express Cloud Service display on the left side of the page. To navigate to the Service Details for your service, see Access Service Details.

- **Billing Metrics**
  
  For non-metered cloud accounts, this metric shows the prorated amount used of your entitlement per day. This number does not affect the amount you are billed for your subscription.

  For metered cloud accounts, this metric shows the number of hours your cloud databases have been running. These numbers are used in billing for the cloud services that are consumed.

  For non-metered cloud accounts, this metric also lists your Database instance quota and how much of it you have used. The quota is utilized when you provision a Cloud database (see Create a Service Instance) and released when you delete a Cloud database (see Delete a Service Instance).

  Alert Rules are the ability to get an email when the quota has reached a particular number.

- **Business Metrics**
This is your service usage metric. You can customize the graphs on this page. The metrics provided on this page include:

- **Object Count** – This is the number of objects in the PDB.
- **Table Count** – This is the number of tables in the PDB.
- **Storage Used (GB)** – This is the amount in GB of storage that is in use.
- **Storage Percent (%)** – This is the percentage of the storage limit that is in use.
- **Storage Limit (GB)** – This is the total amount of storage allocated for this PDB. Because there is some storage overhead required to create the PDB, you are given more storage than the amount you purchased.

- **Documents**
  This option is not available at this time.

- **Status**
  This is the up time of your service including planned outages.

**Related Topics**

- **Use the Cloud Portal to Monitor Database Performance**
  You can monitor the performance of your database in Oracle Database Exadata Express Cloud Service from the My Services page.

## Create a Service Instance

When you create an Oracle Database Exadata Express Cloud Service instance, you provision a Pluggable Database (PDB) that hosts your Exadata Express. To create an Exadata Express instance, use the Create New Oracle Database Exadata Express Cloud Service Instance wizard as described in the following procedure.

To create an Exadata Express instance:

1. Open the My Services dashboard. See [Access My Services](#).
   The My Services dashboard displays.

2. Click **Create Instance**.
   The drop down of service instance choices you have purchased displays.

3. From the drop down, select **Exadata Express**.
   The Create New Service Instance page displays.
4. On the Instance Details page, specify the following:
   - **Name**: Specify a unique name for your service instance. If you specify a name that already exists, the system displays an error and the instance is not created.
   - **Database Shape**: Specify the Database shape that you want to create. This determines the hardware resources provided to your Cloud database, particularly the maximum storage available. The available options are:
     - X20 (1 OCPU, 20 GB storage)
     - X250 (1 OCPU, 250 GB storage)
     - X500 (2 OCPUs, 500 GB storage)
     - X1000 (4 OCPUs, 1 TB storage)
     - X1000IM (4 OCPUs, 1 TB storage, IMC)
   For non-metered cloud accounts, only shapes that still have quota available are displayed. See “Resource Quotas” in Understand Service Details and Metrics. For details about all resources available in each database shape, see Resource Restrictions.
   - **Email**: Enter an email for the Exadata Express administrator.
   - **User Name**: Enter a user name for the Exadata Express instance administrator. Alternatively, check the **Use email as user name** option to copy the Email entry into the User Name field.
   - **First Name**: Enter the first name of the Exadata Express instance administrator.
   - **Last Name**: Enter the last name of the Exadata Express instance administrator.

5. Click **Next**.
   The Confirmation page appears.

6. Review your selections shown on the Confirmation page, click **Create Service Instance**.
   Your new instance is created and appears on the My Services dashboard.
Delete a Service Instance

When you delete a service instance for Oracle Database Exadata Express Cloud Service, all data objects and data is removed. Removing an instance provides you with the ability to completely start over with a new service instance.

To delete an Exadata Express instance:

1. Open the Service Details for the instance you want to delete. See Access Service Details.

   The Service Details display.

2. Click the menu icon next to Open Service Console.

   The menu options display.

3. Select Delete.

   The Delete Instance Confirmation page displays.

4. To confirm, click Delete.

   The PDB instance and all its data objects and data has been removed. It is not possible to recover objects or data once this is done.
Lock a Service Instance

When you lock a service instance for Oracle Database Exadata Express Cloud Service, all access to your service including Oracle Application Express, REST services, and Oracle Net Services (SQL*Net) is disabled. A service user can access the Service Console, but cannot perform operations on this service instance.

**Note:**

For metered cloud accounts, locking a service instance does not pause the meter. Your cloud account continues to be charged at the same rate as the unlocked service instance. To stop the meter, export your data and applications, and delete a service instance.

To lock an Exadata Express instance:

1. Go the Service Details for the instance you want to lock. See Access Service Details.
   The Service Details display.

2. Click the menu icon next to Open Service Console.
   The menu options display.

3. Select **Lock**.
   The Lock Instance Confirmation page displays.

4. To confirm, click **Lock**.
Connect Tools and Utilities

Exadata Express supports these tools and utilities.

Topics:

• About Connecting Tools and Utilities
• Connect SQL*Plus
• Connect SQLcl
• Connect SQL Developer

About Connecting Tools and Utilities

Oracle makes a variety of tools available for use with Oracle Database Exadata Express Cloud Service. Download links to these tools and utilities are displayed in the Exadata Express Service Console. Most of these links point to download pages hosted on Oracle Technology Network (OTN).

You can use Exadata Express with Oracle SQL Developer, an integrated environment for SQL and PL/SQL development, and Oracle SQLcl, an enhanced command-line interface. These tools have been refreshed and now include enhancements specifically for Exadata Express. Please use Oracle SQL Developer version 4.1.5 or later, which is available for download from SQL Developer Downloads on OTN. For Oracle SQLcl, download the most recent version from Oracle SQLcl Downloads on OTN. Note these both require Java to be installed on your machine. You should be using a recently updated Java version plus JCE cryptographic extensions. See Connect SQL Developer for more information about setting up SQL Developer and Connect SQLcl for more information about setting up SQLcl for use with Exadata Express.

Connect SQL*Plus

You can connect to your Oracle Database Exadata Express Cloud Service with SQL*Plus from an Oracle Database Client or an Oracle Instant Client.

Topics

• Connect Using SQL*Plus from Oracle Instant Client
• Connect Using SQL*Plus from Oracle Database Client

Connect Using SQL*Plus from Oracle Instant Client

These instructions explain how to install Oracle Instant Client and connect to the Oracle Database Exadata Express Cloud Service using SQL*Plus. These instructions are also valid for any OCI based driver that uses Oracle Instant Client.

To install, configure and connect:
1. Download the **Client Credentials** zip file from the Exadata Express service console. See *Enable Oracle Net Services (SQL*Net) Access for Client Applications*.

2. Obtain SQL*Plus Instant Client by downloading the **sqlplus** and **basic** packages for the desired architecture from **Instant Client Downloads**. The various options include Microsoft Windows x64 or Windows 32-bit, Linux x86-64, Solaris x86-64, and Solaris SPARC 64-bit.

   **Note:**
   
   The smaller **basiclite** package may be used instead of **basic**. The smaller package contains only English error messages and has support for Unicode, ASCII, and Western European character sets.

3. Unzip the downloaded package into a single directory, for example to C:\instantclient_12_1 or /home/myuser/instantclient_12_1.

4. Set your environment's library loading path to the directory created in Step 3. If you are on a Linux client, set **LD_LIBRARY_PATH** and if you are on a Windows client, set the **PATH** variable.
   
   **a.** On non-Windows platforms, create a symbolic link as:
   
   ```
   cd /home/myuser/instantclient_12_1
   ln -s libclntsh.so.12.1 libclntsh.so
   ```
   
   and set the library and executable paths as:
   
   ```
   export LD_LIBRARY_PATH=/home/myuser/instantclient_12_1:
   instantclient_12_1:$LD_LIBRARY_PATH
   export PATH=/home/myuser/instantclient_12_1:$PATH
   ```
   
   **Note:**
   
   Instead of setting **LD_LIBRARY_PATH**, you can create an `ldconfig` configuration file, if there is no other Oracle software that will be impacted.

   **b.** If you are on a Windows client, append C:\instantclient_12_1 to the **PATH** variable in the "System variables" section of the Environment Variables pane.

5. Unzip the downloaded **Client Credentials** zip file into a directory. For example, move the files to C:\netadmin or /home/myuser/netadmin directory.
Secure the files so that only users who are authorized to connect to the Exadata Express service have access to them. By default, Oracle recommends granting permissions only to the owner of the files.

6. Edit the unzipped sqlnet.ora file, and change the WALLET_LOCATION to the unzipped directory.

   For example in Linux, change

   ```
   WALLET_LOCATION = (SOURCE = (METHOD = file)
   (METHOD_DATA = (DIRECTORY=/?/network/admin)))
   ```

   to

   ```
   WALLET_LOCATION = (SOURCE = (METHOD = file)
   (METHOD_DATA = (DIRECTORY="/home/myuser/netadmin")))
   ```

7. Set the TNS_ADMIN environment variable to the directory with the unzipped files.
   a. If you are on a non-Windows client, set the TNS_ADMIN environment variable to the directory with the unzipped files. For example, in the shell you intend to run your application in, execute `export TNS_ADMIN=/home/myuser/netadmin`
   b. If you are on a Windows client, create a new user variable TNS_ADMIN in the Environment Variables pane and set it to the directory containing the unzipped files, for example `C:\netadmin`.

Note:

This command can also be added to login files such as `HOME/.bashrc`

8. Start SQL*Plus and connect to your service.

   For example if the user ‘CJ’ had been granted access to the database, then run SQL*Plus in a terminal as, `sqlplus cj@dbaccess`

Connect Using SQL*Plus from Oracle Database Client

These instructions explain how to install Oracle Database Client and connect to the Oracle Database Exadata Express Cloud Service using SQL*Plus. These instructions are also valid for any OCI based driver that use Oracle Database Client.

To install, configure and connect:


2. Install Oracle Database Client 12.1.0.2 from Oracle Database Software Downloads on the Oracle Technology Network.
3. Unzip the downloaded Client Credentials zip file into ORACLE_HOME/network/admin subdirectory.

4. If you are on a Windows client, change the DIRECTORY entry in SQLNET.ORA by replacing forward slash (/) with back slash (\).

5. Start SQL*Plus and connect to your service.
   
   For example: sqlplus pdb_admin@dbaccess
   Enter password: password

Connect SQLcl

SQLcl is a new enhanced SQL command-line implemented in Java. Your Oracle Database Exadata Express Cloud Service supports connecting to your PDB from SQLcl.

Topics

• Download and Set up Oracle SQLcl
• Configure Oracle SQLcl Cloud Connection
• Invoke SQLcl from Oracle Developer Cloud Service

Download and Set up Oracle SQLcl

Oracle SQLcl provides a powerful command-line interface (CLI) for authoring SQL and PL/SQL on databases. This section describes the download requirements and prerequisites for Oracle SQLcl and where to find the latest setup instructions.

Requirements and Prerequisites

• The functionality described here uses the Oracle SQL Developer Command-Line (SQLcl), version 17.4 (current version) or later running on Java Runtime Environment (JRE) version 8 update 161 or later.

• Network connections between SQLcl and Exadata Express require Java unlimited strength encryption. Verify that SQLcl is running on JRE version 8 update 161 or later, which provides unlimited strength encryption by default.

Download and Setup

• For download and setup instructions to set up Oracle SQLcl locally, see Oracle SQLcl documentation on the Oracle Technology Network: SQL Developer DownloadsSQL Developer Command Line SQLcl Overview
Configure Oracle SQLcl Cloud Connection

Oracle SQLcl can be configured to connect to Oracle Database Exadata Express Cloud Service, as a regular database connection using the client credentials downloaded from the Exadata Express service console.

To create an Oracle SQLcl cloud connection:

1. Download the Client Credentials zip file from the Exadata Express service console. See Enable Oracle Net Services (SQL*Net) Access for Client Applications for information on downloading the Client Credentials.

   **Note:**
   Copy the Client Credentials zip file into ~/sqlcl/bin directory, to avoid including the complete file path in step 3 below.

2. Navigate to the sqlcl/bin directory from where you unzipped the SQLcl installation files, and run sql /nolog, to startup Oracle SQLcl.

   The Oracle SQLcl starts displaying the date and time, the SQLcl version and copyright information, before the SQLcl prompt appears.

   ```
   D:\PDB Service\SQL CL\sqlcl-no-jre-latest\sqlcl\bin>sql /nolog
   SQLcl: Release 17.4.0.16.160.2007 RC on Fri Dec 08 12:18:07 2017
   Copyright (c) 1982, 2017, Oracle. All rights reserved.
   SQL>
   ```

3. At the SQLcl prompt, type `set cloudconfig <name of your wallet zip file>`, and press the Enter key.

   **Note:**
   Include the complete file path, if you have not copied the Client Credentials zip file to the ~/sqlcl/bin directory, earlier.

4. Enter the Password provided for downloading the Client Credentials from the Exadata Express service console, and press the Enter key.

   ```
   SQL> set cloudconfig client_credentials.zip
   Wallet Password: ************
   ```

   Using temp directory:C:\Users\USER~1.ORA\AppData\Local\Temp
   \oracle_cloud_confi
g6707346342028726502
5. To connect to the Exadata Express, type `conn <username>/
<password>@<servicename>`, and press the Enter key. The <username> and password are the credentials of your database account.

You should now be connected to Exadata Express.

```
SQL> conn pdb_admin/password@dbaccess
Connected.SQL>
```

**Note:**
The example above uses the administrator user credentials to connect to the database. These are specified as part of the service creation in the cloud portal.

**Invoke SQLcl from Oracle Developer Cloud Service**

You can configure build steps in Oracle Developer Cloud Service to invoke SQL statements using SQLcl, a powerful command-line utility for managing and querying Oracle Database. The Invoke SQLcl build step now allows you to connect to Oracle Database Exadata Express Cloud Service.

Learn more about Oracle Developer Cloud Service. To invoke SQLcl and connect to Oracle Database Exadata Express Cloud Service, see Adding a Build Step that Invokes SQLcl in Using Oracle Developer Cloud Service.

**Related Topics**
- Download Client Credentials
- Oracle SQL Developer Command Line

**Connect SQL Developer**

To connect Oracle SQL Developer to your Oracle Database Exadata Express Cloud Service, you need to generate a secure wallet by enabling Oracle Net Services (SQL*Net) for your service, install Oracle SQL Developer, and create a cloud connection from Oracle SQL Developer to your service.

**Topics**
- Install Oracle SQL Developer
- Configure Oracle SQL Developer Cloud Connection

**Install Oracle SQL Developer**

Oracle SQL Developer is a free integrated development environment that simplifies the development and management of Oracle Database in both traditional and Cloud deployments. Oracle SQL Developer version 17.4 (current version) or later, can connect to Oracle Database Exadata Express Cloud Service using Oracle Wallet,
which can be downloaded from the database service. This section describes how to install Oracle SQL Developer.

**Note:**

The functionality described here requires the Oracle SQL Developer Release version 17.4 (current version) or later running on Java Development Kit (JDK) version 8 update 161 or later.

For download and setup instructions to set up Oracle SQL Developer locally for Exadata Express, see Using SQL Developer with Oracle Database Exadata Express Cloud Service on Oracle Technology Network.

**Note:**

Network connections between Oracle SQL Developer and Exadata Express require Java unlimited strength encryption. Verify that Oracle SQL Developer is running on JDK version 8 update 161 or later, which provides unlimited strength encryption by default. As a convenience specifically for Windows 64-bit users, Oracle SQL Developer provides an option to install the correct JDK version as part of the base software installation.

### Configure Oracle SQL Developer Cloud Connection

Oracle SQL Developer must have an Oracle Cloud connection configured to connect to Oracle Database Exadata Express Cloud Service. The connection can be created using your Exadata Express database account credentials.

To add an Oracle Cloud connection:

1. Run Oracle SQL Developer locally.

   The Oracle SQL Developer home page displays.
2. Under Connections, right click **Connections**.
The Connection menu appears.

3. Select **New Connection**.
The New/Select Database Connection dialog appears.
4. On the New/Select Database Connection dialog, make the following entries:
   - Connection Name - Enter the name for this cloud connection.
   - Username - Enter the database username. You can either use the default administrator database account (PDB_ADMIN) provided as part of the service or create a new schema, and use it.
   - Password - Enter the Password required during sign in when accessing Exadata Express.
   - Connection Type - Select Cloud PDB.
   - Configuration File - Click Browse, and select the Client Credentials zip file, downloaded from the Exadata Express service console. See Enable Oracle Net Services (SQL*Net) Access for Client Applications for information on downloading the Client Credentials.
   - Keystore Password - Enter the Password generated while downloading the Client Credentials from the Exadata Express service console.

5. Click Test.
Status: Success displays at the left most bottom of the New/Select Database Connection dialog.

6. Click **Connect**.
   An entry for the new connection appears under Connections.

7. Open the new connection.
   If you have connected successfully, the tables and other objects from Exadata Express display under the new connection.
Connect Apps, Clients and Drivers

Oracle Database Exadata Express Cloud Service supports connecting to your service from a variety of database applications, clients and drivers over Oracle Net Services (SQL*Net).

Topics:
• About Database Clients and Versions
• Database Client Prerequisites
• Connect with JDBC Thin Driver and UCP
• Connect with Microsoft .NET and Visual Studio
• Connect with Node.js

About Database Clients and Versions

Oracle makes a variety of database client drivers available for use with Oracle Database Exadata Express Cloud Service.

Download links to these clients are displayed in the Exadata Express Service Console. Most of these links point to download pages hosted on the Oracle Technology Network (OTN). In almost all cases, database client versions supported with Exadata Express include Oracle Database 12c Release 1 (12.1.0.2) and Oracle Database 12c Release 2 (12.2.0.1).

Database Client Prerequisites

This section describes the common configuration tasks you need to perform before a database client can communicate with the Oracle Database Exadata Express Cloud Service.

Topics:
• About Database Client Prerequisites
• Enable Oracle Net Services (SQL*Net) Access for Client Applications
• Download Client Credentials

About Database Client Prerequisites

Database clients and the Oracle Database Exadata Express Cloud Service must be configured to communicate with each other. These prerequisite configurations include enabling Oracle Net Services (SQL*Net) for your service, downloading client credentials, and following the specific installation and set-up instructions for the connecting database client.

Prerequisite tasks for database client connectivity:
• Enable Oracle Net Services (SQL*Net) access to your service. See Enable Oracle Net Services (SQL*Net) Access for Client Applications.

• Download client credentials. See Download Client Credentials.

• Follow set-up instructions for the specific database client you want to connect with. See one of the following for instructions:
  – SQLcl. See Connect SQLcl.
  – SQL Developer. See Connect SQL Developer.
  – JDBC Thin Client. See Property Settings for JDBC Thin Driver and UCP.
  – Node.js. See Connect with Node.js.

• Please review the recommendations outlined in Simultaneous Connection Considerations.

### Enable Oracle Net Services (SQL*Net) Access for Client Applications

The Oracle Database Exadata Express Cloud Service supports connecting to your service from diverse database clients over Oracle Net Services, also known as SQL*Net. Some examples of supported clients include SQL*Plus, SQLcl, SQL Developer, JDBC Thin, ODP.NET, OCI, and Instant Client. Database drivers for popular programming and scripting languages that are supported include Python, PHP, Node.js, C/C++, Ruby and Perl. Enabling Oracle Net Services (SQL*Net) access is a prerequisite for all clients and drivers connecting over Oracle Net Services.

**Note:**

Client connections to your Exadata Express cloud database automatically disconnects after 30 minutes of idle time. This timeout interval cannot be changed. To keep your client connections alive, make sure they execute an operation on the database at least once during each 30-minute period. See Feature Restrictions and Limitations in Known Issues for Oracle Database Exadata Express Cloud Service.

The Service Administrator must do the following to enable Oracle Net Services (SQL*Net):

1. Open the Develop page of the service console for Exadata Express. See Access the Service Console.

2. Under Client Development Setup section, click Client Access.

3. On the pop-up window, click Enable Client Access.

4. Download the client credentials. See Download Client Credentials.

5. Now, depending on the client-side application and driver being used, you need to configure the application connection string for that application. See Connect Apps, Clients and Drivers.
Download Client Credentials

The Oracle Database Exadata Express Cloud Service service console provides an easy way to download client credentials.

The downloaded zip file contains important information required to make encrypted network connections. Its contents include Oracle Wallet and Java Keystore (both contain a client certificate) as well as essential client configuration files. Two of these files are password protected. When making a cloud database connection, your client will read the credentials zip file, and under certain conditions, prompt for the password you entered upon downloading. Finally, your client will authenticate to the cloud database using a separate database user name and password you will be asked to provide.

Note that client credential zip files should be carefully managed. Please remember to keep the file secure to avoid unauthorized database access. You may wish to share this file with other authorized users who need to connect to your cloud database. If you believe the security of this file has been compromised, then immediately disable client access using the cloud service console. Disabling client access will invalidate existing client credentials, effectively shutting off network access to your database. Later, you can re-enable client access and download a new credentials zip file. If you accidentally delete your credentials zip file, then you can download a new one using the cloud service console without invalidating existing clients.

1. Navigate to the service console for Exadata Express and open the Develop page. See Access the Service Console

2. Click Client Credentials to download a zip file containing your security credentials and network configuration files.

3. Enter a password to create a password-protected Oracle Wallet and Java Keystore files for the service.
4. Save the downloaded zip file to a secure location that is accessible by your database client(s).

5. Now, depending on the client-side application and driver being used, you may need to extract certain contents of the downloaded zip file and/or configure the connection string for your client application. See Connect Apps, Clients and Drivers, about setting up your specific client.

Connect with JDBC Thin Driver and UCP

This section describes the software prerequisites and property settings required for connecting to Oracle Database Exadata Express Cloud Service using JDBC thin driver.

Topics

• Software Requirements for JDBC Thin and UCP Connections
• Property Settings for JDBC Thin Driver and UCP

Software Requirements for JDBC Thin and UCP Connections

Before connecting with JDBC Thin driver to the Oracle Database Exadata Express Cloud Service, you must satisfy the following software requirements.

• You must have JDK 8 (any version JDK8u71 or higher) or JDK 7 (JDK7u80).
• Download the JCE Unlimited Strength Jurisdiction Policy Files from here. Follow installation notes in the README.
• Download the 12.1.0.2 JDBC Thin driver (ojdbc7.jar) and UCP (ucp.jar) from Oracle Technology Network.

Note:

You must use a patched version of 12.1.0.2 JDBC Thin driver (ojdbc7.jar) from Oracle Technology Cloud Download.

See JDBC/UCP Connectivity for Java SE, Java EE Containers, and Java Cloud Service and JDBC/UCP Connectivity for Java IDEs for more details.

Property Settings for JDBC Thin Driver and UCP

Make sure you have satisfied software prerequisites and property setting requirements, before connecting with JDBC Thin driver and UCP to Oracle Database Exadata Express Cloud Service.

Note:

These properties can be set as system properties or connection properties. For further details on setting these as connection level properties, see Oracle Database JDBC Java API Reference 12c Release 2(12.2.0.1).
To set properties as system properties:


2. Make sure to place all required configuration files tnsnames.ora, truststore.jks, and keystore.jks (created while enabling Oracle Net Services (SQL*Net) access for client applications) to an accessible location. These credentials are downloaded by clicking Download Client Credentials from the Service Console. See Download Client Credentials

3. Set the path of tnsnames.ora file using oracle.net.tns_admin system property.

4. Enable the server DN match by setting oracle.net.ssl_server_dn_match system property to true.

5. Set the location for keystore and truststore JKS files, using javax.net.ssl.trustStore and javax.net.ssl.keyStore system properties.

6. Set the password for JKS using javax.net.ssl.keyStorePassword and javax.net.ssl.trustStorePassword. Use the password provided while downloading the client credentials.

7. Set the ssl property by setting oracle.net.ssl_version to 1.2.

8. If using JDK7, set

   oracle.net.ssl_cipher_suites=(TLS_RSA_WITH_AES_256_CBC_SHA256)

Using 12.1.0.2 JDBC Thin Driver with JDK8:

```
java -Doracle.net.tns_admin=/home/user1/cloud
     -Djavax.net.ssl.trustStore=truststore.jks
     -Djavax.net.ssl.trustStorePassword=password
     -Djavax.net.ssl.keyStore=keystore.jks
     -Djavax.net.ssl.keyStorePassword=password
     -Doracle.net.ssl_server_dn_match=true
     -Doracle.net.ssl_version=1.2
     SelectTest
```

Using 12.1.0.2 JDBC Thin Driver with JDK7:

```
java -Doracle.net.tns_admin=/home/user1/cloud
     -Djavax.net.ssl.trustStore=truststore.jks
     -Djavax.net.ssl.trustStorePassword=password
     -Djavax.net.ssl.keyStore=keystore.jks
     -Djavax.net.ssl.keyStorePassword=password
     -Doracle.net.ssl_server_dn_match=true
     -Doracle.net.ssl_version=1.2
     -Doracle.net.ssl_cipher_suites=TLS_RSA_WITH_AES_256_CBC_SHA256
     SelectTest
```

Replace “password” in the above examples with your wallet password provided during wallet download.

Sample code for UCPSample.java:

```java
import java.sql.Connection;
import java.sql.DatabaseMetaData;
```
import java.sql.SQLException;
import java.sql.ResultSet;
import java.sql.Statement;
import oracle.ucp.jdbc.PoolDataSourceFactory;
import oracle.ucp.jdbc.PoolDataSource;

public class UCPSample {
    // Connection URL. 'dbaccess' is from 'tnsnames.ora' file
    final static String DB_URL = "jdbc:oracle:thin:@dbaccess";
    final static String DB_USER = "pdb_admin";
    final static String DB_PASSWORD = "password";
    final static String CONN_FACTORY_CLASS_NAME = "oracle.jdbc.pool.OracleDataSource";

    public static void main(String args[]) throws SQLException {
        // Get the PoolDataSource for UCP
        PoolDataSource pds = PoolDataSourceFactory.getPoolDataSource();
        // Set the connection factory first before all other properties
        pds.setConnectionFactoryClassName(CONN_FACTORY_CLASS_NAME);
        pds.setUrl(DB_URL);
        pds.setUser(DB_USER);
        pds.setPassword(DB_PASSWORD);
        pds.setConnectionPoolName("JDBC_UCP_POOL");

        // Set pool properties
        pds.setInitialPoolSize(5);
        pds.setMinPoolSize(5);
        pds.setMaxPoolSize(20);

        // Get a connection to the cloud database
        try (Connection connection = pds.getConnection()) {
            // Get the JDBC driver name and version
            DatabaseMetaData dbmd = connection.getMetaData();
            System.out.print("Driver Name: " + dbmd.getDriverName());
            System.out.print("Driver Version: " + dbmd.getDriverVersion());

            // Statement and ResultSet are AutoCloseable and closed automatically.
            try (Statement statement = connection.createStatement()) {
                try (ResultSet resultSet = statement.executeQuery("select sysdate from dual")) {
                    while (resultSet.next())
                        System.out.println("Today's Date is: " + resultSet.getString(1));
                }
                System.out.print("Successfully connected to a cloud database");
            }
        }
    }
}

See JDBC/UCP Connectivity for Java SE, Java EE Containers, and Java Cloud Service and JDBC/UCP Connectivity for Java IDEs for more details.
Connect with Microsoft .NET and Visual Studio

Oracle Database Exadata Express Cloud Service supports connectivity to the Microsoft .NET Framework and Visual Studio.

Oracle Data Provider for .NET, Unmanaged Driver and Oracle Data Provider for .NET, Managed Driver provide run-time ADO.NET data access to the database service. Oracle Developer Tools for Visual Studio support design-time support for the database service. Oracle Providers for ASP.NET can store web application state using the service. These software components are bundled together in a single software installation called Oracle Data Access Components (ODAC) for Windows and is a free download.

Set-up Instructions

Oracle recommends using the latest ODAC version with Exadata Express. Click the following link for instructions on how to download, install and configure ODAC:
Oracle Technology Network ODAC Setup Instructions for Oracle Cloud

Connect with Node.js

This section explains how to connect an app using Node.js to your cloud service.

Topics:
• Connect an App with Node.js
• Connect with Node.js in ACCS

Connect an App with Node.js

Node.js is an open source, cross-platform runtime environment for writing mid-tier and networking applications in JavaScript. This Oracle Database driver can be used to connect to Oracle Database Exadata Express Cloud Service using Instant Client 12.1 with security enforced by using an Oracle Wallet.

In order to connect with Node.js, you need to:
• Download and install Instant Client – see Oracle Instant Client
• Enable SQL*Net for your service – see Enable Oracle Net Services (SQL*Net) Access for Client Applications
• Download client credentials – see Download Client Credentials
• Install node-oracledb for Node.js database driver – see Installing node-oracledb

For additional resources, see Node.js Developer Center on the Oracle Technology Network (OTN).

Enable Oracle Network Connectivity and Download the Client Wallet:

1. If Client Access is not already enabled, follow steps outlined in Enable Oracle Net Services (SQL*Net) Access for Client Applications.
2. If not already downloaded, follow steps in Download Client Credentials to download security credentials and network configuration files that allow client access to your Exadata Express database.

3. Unzip the client credentials and move the contents to a directory. For example move the files to

   C:\netadmin or /home/myuser/netadmin

Secure the files so that only users who are authorized to connect to your Exadata Express service have access to them. By default, Oracle recommends granting permissions only to the owner of the files.

4. Edit the unzipped sqlnet.ora file and change the wallet location to the unzipped directory.

   For example, change:

   
   ```
   WALLET_LOCATION = (SOURCE = (METHOD = file)
   (METHOD_DATA = (DIRECTORY=?/network/admin)))
   ```

   to:

   ```
   WALLET_LOCATION = (SOURCE = (METHOD = file)
   (METHOD_DATA = (DIRECTORY="C:\netadmin")))
   ```

   or, on Linux, to:

   ```
   WALLET_LOCATION = (SOURCE = (METHOD = file)
   (METHOD_DATA = (DIRECTORY="/home/myuser/netadmin")))
   ```

5. On Windows, create a new user variable TNS_ADMIN in the Environment Variables pane and set it to the directory containing the unzipped files, for example C:\netadmin. On Windows 8 this pane can be accessed by navigating to Control Panel>System>Advanced System Settings>Environment Variables.

6. On non-Windows platforms, set the TNS_ADMIN environment variable to the directory with the unzipped files.

   For example, in the shell you intend to run your application in, execute:

   ```
   export TNS_ADMIN=/home/myuser/netadmin
   ```

   This command can also be added to login files such as $HOME/.bashrc

**Install Node.js Database Driver**

1. Install node-oracledb for Node.js using the updated Instant Client. The steps for installing node-oracledb for Node.js are specific to your platform and environment. For installation steps for various platforms and environments, see the Installing node-oracledb manual.

2. For additional user documentation, see Documentation for the Oracle Database Node.js Add-on.
Run Your Applications

1. Update your application to use the Oracle Network Service name given in the unzipped `tnsnames.ora` file. For example, use `dbaccess` as the connect string.

   Alternatively, change the connect string in `tnsnames.ora` to match the string used by your application.

2. Review Known Issues for Oracle Database Exadata Express Cloud Service

3. Run your application.

Connect with Node.js in ACCS

Node.js is an open source, cross-platform runtime environment for writing mid-tier and networking applications in JavaScript. Oracle Application Container Cloud Service (ACCS) makes it easy to deploy container-based applications in on a variety of platforms, including Node.js.

The following steps demonstrate how to connect Exadata Express from ACCS.

Pre-requisites

You must have an Oracle Cloud account with Exadata Express and ACCS.

Create a Test Application

In this step, you will create a Node.js application to be deployed to ACCS. The application will run a basic connection test and use a simple web server to report the results of the test.

1. Create a new directory named `connection-test-app`.
2. Add a file named `manifest.json` to `connection-test-app` with the following contents:

   ```json
   {
   "runtime":{
   "majorVersion":"6"
   },
   "command": "node index.js",
   "release": {},
   "notes": ""
   }
   ```

   ACCS applications often have one or two metadata files. In this case, the `manifest.json` specifies the version of Node.js to run and the command that will be used to start the application.

3. Add a file named `index.js` to `connection-test-app` with the following contents:

   ```javascript
   const http = require('http');
   const oracledb = require('oracledb');
   let error;
   let user;
   ```
oracledb.getConnection(
    {
        user: process.env.EECS_USER,
        password: process.env.EECS_PASSWORD,
        connectString: 'dbaccess'
    },
    function(err, connection) {
        if (err) {error = err; return;}

        connection.execute('select user from dual', [], function(err, result) {
            if (err) {error = err; return;}

            user = result.rows[0][0];
            error = null;

            connection.close(function(err) {
                if (err) {console.log(err);}  
            });
        });
    }
);

http.createServer(function(request, response) {
    response.writeHead(200, {'Content-Type': 'text/plain'});

    if (error === null) {
        response.end('Connection test succeeded. You connected to Exadata Express as ' + user + '!');
    } else if (error instanceof Error) {
        response.write('Connection test failed. Check the settings and redeploy app!
');
        response.end(error.message);
    } else {
        response.end('Connection test pending. Refresh after a few seconds...');
    }
}).listen(process.env.PORT);

---

**Note:**

Three environment variables are referenced through `process.env`. `PORT` is defined by ACCS, but the other environment variables will be created in the last step of this procedure.

---

**Add Client Credentials**

In this step, you will download the client credentials to connect to Exadata Express and place them with the application files.
1. Download the client credentials to connect to Exadata Express database. See Download Client Credentials. Once downloaded, you must treat the files securely to prevent unauthorized database access.

2. Extract the contents of the client_credentials.zip file to the connection-test-app directory created earlier.

3. Change directories into the client_credentials directory and open the sqlnet.ora file in a text editor. sqlnet.ora file and change the value of the directory from:

   ?/network/admin

   to

   /u01/app/client_credentials.

   With ACCS, application files are copied to /u01/app/ directory of the container. The sqlnet.ora file must point to the location of the client credentials, which can differ depending on the environment.

Deploy the Application

At this point the application can be deployed to ACCS.

1. To deploy the application to ACCS, change the directories back up to extracted directory and compress the contents in a new zip file.

2. Return to the browser and navigate to the ACCS service console. Click Create Application.

3. Select Node as the application platform.

4. Provide an application name.

5. Use the Archive file picker to select the application archive created earlier.
6. Set Instances and Memory (GB): to 1.

7. Click Create to start the deployment. The application appears in the list of ACCS applications where you can obtain the URL. Navigating to the URL at this point will show the test has failed, this is expected.

8. Return to the ACCS service console and expand the application you just added.

9. Click Deployments on the left navigation and click Add under Environmental Variables.
10. Set **Name** to **TNS_ADMIN** and **Value** as **$APP_HOME/client_credentials** and click **Save**.

**Note:**
If you are developing locally, the **TNS_ADMIN** environment variable can point to a local copy of the client credentials used for development.

11. Repeat Step 10 to create two more environmental variables. The first will be named as **EECS_USER** and its value will be the username of Exadata Express database user you want to connect with. The last environment variable will be named **EECS_PASSWORD** and its value will be the password of the database user specified in **EECS_USER**.

12. Deploy these settings by clicking **Apply Edits**. When the application finishes deploying, you can navigate to its URL again and you will see that the connection test has succeeded.
Develop Document Store Applications Using SODA

Oracle Database 12c stores, manages, and indexes JSON data. Application developers can work with JSON, as well as any other type of data, using NoSQL style APIs called Simple Oracle Document Access (SODA). Under the covers, SODA stores the data in regular Oracle Database tables, so SQL access using standard tools is still allowed.

Topics

- About Using the Oracle Database as a Document Store through SODA
- Enable SODA for Your Service
- Use SODA for REST
- Use SODA for Java
- Use SODA from SQLcl

About Using the Oracle Database as a Document Store through SODA

Simple Oracle Document Access (SODA) is a family of APIs that provides a very easy and intuitive way to persist JavaScript Object Notation (JSON) documents in the Oracle database. Operations on the documents supported by SODA include create, remove, update, and delete (CRUD).

SODA allows the Oracle Database to be used as a powerful NoSQL store, supporting key-based access to all documents, and query-based access to JSON documents, all without needing to use SQL. Because SODA is built on top of the Oracle database, you get proven Oracle enterprise-grade reliability and many features typically lacking in NoSQL stores, such as transactions. If desired, you can still access SODA documents directly with SQL.

The primary abstraction provided by SODA is a collection of documents. SODA is particularly powerful when it comes to JSON documents, though all other types of documents are supported. JSON documents in SODA can be queried using intuitive template-like queries, without needing to write SQL. These queries are also expressed in JSON, and called QBEs (query-by-example).

SODA is built on top of native JSON support in the Oracle database. Under the covers, document collections are stored in Oracle database relational tables, which can be accessed directly as well, for example by reporting tools.

The Exadata Express service provides the following implementations of SODA:

- SODA Over REST
This implementation of SODA is part of Oracle REST Data Services (ORDS), and is usable from any language capable of making HTTP requests. See About SODA for REST.

- SODA for Java
  The Java library supports SODA for use with the Java programming language. See Use SODA for Java.

- SODA for SQLcl
  SQLcl command line shell supports basic SODA commands and also supports scripting using SODA by allowing SODA Java classes to be invoked from Javascript. See Use SODA from SQLcl.

Enable SODA for Your Service

Before you can use SODA APIs for your Oracle Database Exadata Express Cloud Service, you need to setup a database account with the appropriate grants and tablespace attributes.

There are a couple of ways to enable your service for SODA:

- Using the Service Console to Enable SODA
- Enabling SODA Manually

Using the Service Console to Enable SODA

To enable your service for SODA from the Service Console:

1. Go to the service console for your service and open the Manage page of the service console. See About Manage.
   The Manage page displays.

2. Click Document Store.
   The Create Document Store dialog appears.
3. To specify a schema that already exists to manage your Document Collections:
   a. Select **Existing Schema**.
   b. Select the schema name from the drop down list.

4. To create a new schema to manage your Document Collections:
   a. Select **New Schema**.

   **Note:**
   You can specify the Oracle Database default schema. See Find the Default Schema Name and Set the Password.
b. Enter the schema name in the text box.

c. For Password, enter the schema password.

d. For Confirm Password, re-enter the schema password.

5. To enable REST based operations on the schema, select **Enable SODA for REST**.

6. Click **Create Document Store**.

7. Click **Done**.

**Enabling SODA Manually**

**Note:**

It is recommended that you enable SODA using the Service Console rather than enabling manually. See [Using the Service Console to Enable SODA](#).

The following settings are required for the database account:

- Grant required roles and privileges

In order to enable SODA for a user the service administrator must grant the **SODA_APP** role to that schema. SODA requires the **SODA_APP** role to be granted to the user, which is the schema name, that will be used to work with collections. The underlying tables used to store document collections will be stored in this schema.
To allow SODA interactions, if not already granted, the following grants are also needed:

- CREATE_SESSION
- CREATE_TABLE
- CREATE_SEQUENCE
- CREATE_TRIGGER
- CREATE_PROCEDURE
- CREATE_VIEW

• Automatic Segment Space Management (ASSM)
  The tablespace must have ASSM enabled.

• To enable SODA for REST, execute:

```sql
exec ords.enable_schema;
commit;
```

To manually grant required roles and privileges:

1. Connect with SQL*Plus, SQLcl or SQL Developer as the PDB_ADMIN user. See Connect SQL*Plus, Connect SQLcl, or Connect SQL Developer.

2. In SQL*Plus, issue the following commands where `schemaName` is the name of the schema you are configuring for use with SODA:

```sql
GRANT SODA_APP to schemaName;
GRANT CREATE SESSION to schemaName;
GRANT CREATE TABLE to schemaName;
GRANT CREATE SEQUENCE to schemaName;
GRANT CREATE TRIGGER to schemaName;
GRANT CREATE PROCEDURE to schemaName;
GRANT CREATE VIEW to schemaName;
```

### Use SODA for REST

Oracle Database Exadata Express Cloud Service supports SODA for REST.

**Topics**

- About SODA for REST
- SODA for REST Prerequisites
- Set up Users and Roles for SODA for REST
- Determine the SODA for REST HTTP Operation URIs
- Get Started with SODA for REST Operations

### About SODA for REST

Oracle Database Exadata Express Cloud Service supports Simple Oracle Document Access (SODA) using Representational State Transfer (REST).
SODA for REST can be used from any modern programming language capable of making HTTP requests. For further details including a complete list of SODA for REST HTTP operations available for the SODA for REST API, see REST Data Services SODA for REST Developer's Guide.

SODA for REST Prerequisites

Prerequisites must be satisfied in order to use SODA for REST.

The following must be performed or known before attempting to use SODA for REST:

• You must enable SODA for your service. See Enable SODA for Your Service.
• You must provide the appropriate roles to users using SODA for REST. See Assign User Roles for SODA for REST.
• You must know how to determine the REST URI. See Determine the SODA for REST HTTP Operation URIs.

Set up Users and Roles for SODA for REST

Service users that want to use SODA for REST must be assigned the appropriate role and must use the proper authentication mechanism.

Topics

• Assign User Roles for SODA for REST
• Create Custom Roles
• About Authentication Mechanisms

Assign User Roles for SODA for REST

Before a user can access SODA for REST, users must be assigned the predefined roles of Database Administrator or Database Developer and your service must be enabled for SODA for REST. It is also possible to create custom roles for accessing SODA for REST.

The initial Service Administrator User for the service is automatically assigned the Database Administrator and Database Developer roles. Additional users need to be manually assigned these roles. You can change the roles assigned to users from My Services. Roles can also be assigned during user creation. See Adding Users and Assigning Roles in Getting Started with Oracle Cloud.

To assign Database Administrator and Database Developer roles to a user:

1. Go to My Services. See Access Service Details.
   My Services displays.
2. Click **Users** icon.
   The Users page displays showing all users provisioned for your service.

3. Find the user you want to assign the role to, and click the **menu** icon.
   A drop down list appears.

4. Select **Manage Roles**.
   The Manage Roles dialog display.
5. Under Available Roles, select the **Database Administrator** and **Database Developer** roles you want to assign to the user and click the arrow to move them to Selected Roles.

   All roles you want to assign to this user are listed under Selected Roles.

6. Click **Save**.

Create Custom Roles

In addition to the predefined roles, such as Database Administrator and Database Developer, you can create your own custom roles. A custom role, for example, allows you to guard access to a particular SODA document collection, instead of a whole schema.

The custom role name is created and defined from the My Services Users page. The custom role can then be assigned to users. You need to perform PL/SQL commands to define what the custom role does. See Security in *Oracle REST Data Services SODA for REST Developer's Guide*.

To create your own custom role:

1. Go to Service Details. See Access Service Details.

   Service Details page displays.
2. **Select Users.**

   The Users page displays showing all users provisioned for your service.

3. **Select Custom Roles.**

   The Custom Roles page appears.
4. Click **Add**.

The Add Custom Role dialog display.

5. Fill out the fields in the dialog. The Role Name is the only required field. This name is used later to identify the role when assigning it to a user.

6. Click **Add**.

The Custom Roles page appears and lists the added custom role.

7. Add this new custom role to a user. Follow steps in **Assign User Roles for SODA for REST**, but instead of adding the Database Administrator and Database Developer roles, add the custom role.

8. Now you need to define what this custom role does by creating and executing PL/SQL commands that define the privilege mapping for this custom role. For example, assuming the EmployeeRole custom role has been created and assigned to a user, in the schema that holds the collection, execute this:

```
Note:

Before running this example, you must connect to the schema that holds the collection with SQL*Plus, SQLcl, or SQL Developer.

declare l_patterns owa.vc_arr;
bEGIN
    l_patterns(1) := '/soda/latest/employee';
    l_patterns(2) := '/soda/latest/employee/*';
    ords.create_role('EmployeeRole');
    ords.create_privilege(p_name => 'EmployeePrivilege', p_role_name => 'EmployeeRole');
```

---

**Image:**
- Page 4: Click Add.
- Page 5: Fill out the fields in the dialog. The Role Name is the only required field. This name is used later to identify the role when assigning it to a user.
- Page 6: Click Add.
- Page 7: Add this new custom role to a user. Follow steps in Assign User Roles for SODA for REST, but instead of adding the Database Administrator and Database Developer roles, add the custom role.
- Page 8: Now you need to define what this custom role does by creating and executing PL/SQL commands that define the privilege mapping for this custom role. For example, assuming the EmployeeRole custom role has been created and assigned to a user, in the schema that holds the collection, execute this:
ords.create_privilege_mapping(p_privilege_name => 'EmployeePrivilege', p_patterns => l_patterns);  commit;
end;

This example creates a privilege mapping that ensures only users with this custom role named EmployeeRole, can access the employee collection. For more information about this example, see Security in Oracle REST Data Services SODA for REST Developer’s Guide.

About Authentication Mechanisms

Simple Oracle Document Access (SODA) for REST is built on top of Oracle REST Data Services (ORDS), which provides various authentication mechanisms, such as basic authentication, OAuth, and so on.

See Authentication Mechanisms in Oracle REST Data Services SODA for REST Developer’s Guide.

Determine the SODA for REST HTTP Operation URIs

A SODA for REST HTTP URI includes the host name of your Oracle Database Exadata Express Cloud Service and, for non-default schemas, the schema name. The default schema name is the name of the schema your service was provisioned with and is not included in the URI.

A SODA for REST HTTP Operation has a Universal Resource Identifier (URI) of this form:

https://hostname/apex/schema(only if non-default schema)/soda/latest

where:

- hostname is the host name found in the Service Instance URL displayed on the Service Details page.
- schema is the name of the non default schema. For the default schema the schema name is not included.

Example URL Pattern for Default Schema:

https://myhostname/apex/soda/latest

Example URL Pattern for Non-default Schema:

https://myhostname/apex/myschemaname/soda/latest

To find the host name:

1. Go to the Service Details page for your service. See Access Service Details.
2. On the Service Details page, locate the Service Instance URL. The host name is the first portion of the URL after https:// and before the subsequent /.
Get Started with SODA for REST Operations

Some basic SODA operations using REST are provided to demonstrate the steps needed to list all collections, create a new collection, insert a document, retrieve a document, and finally to delete the created collections that allow documents to be stored in, and retrieved from, document collections.

For further details on the example operations described below, see Getting Started with SODA for REST and SODA for REST HTTP Operations in Oracle REST Data Services SODA for REST Developer’s Guide.

To perform some simple SODA for REST operations using cURL a common command-line tool:

```
Note:
The following example operations use Basic Authentication with a fictitious user user@example.com with password password and the user has a role of either Database Administrator, Database Developer or both. It is not secure to specify usernames and passwords in clear text as part of the cURL command. Although, for the purposes of brevity, the sample commands shown here use clear text for the username and password.
```

1. Make sure SODA for REST prerequisites have been satisfied. See SODA for REST Prerequisites.
2. To list all collections in the default schema, run this command:

```bash
curl -i -X GET -u user@example.com:password https://exadata1.oraclecorp.com/apex/soda/latest
```

Command results assuming no existing collections in the default schema exist:

```json
dict(items=[], more=False)
```

3. To create a new collection named **MyCollection**, run this command:

The curl option `-i` in the following command is used to include output headers in the response.

```bash
curl -i -X PUT -u user@example.com:password https://exadata1.oraclecorp.com/apex/soda/latest/MyCollection
```

Command result contains:

```plaintext
... HTTP/1.1 201 Created ...
```

4. To list all collections in the default schema again, run this command:

```bash
curl -i -X GET -u user@example.com:password https://exadata1.oraclecorp.com/apex/soda/latest
```

Command result includes:

The following JSON response shows the default schema contains the **MyCollection** collection created in a previous step. Metadata for this collection is shown under "properties" as part of the returned JSON (this part of the output is abridged for brevity).

```json
dict(items=[dict(name='MyCollection', properties=dict(schemaName='BXJHVBGSPT', tableName='MyCollection', keyColumn=dict(name='ID', ... ))), links=[dict(rel='canonical', href='http://exadata1.oraclecorp.com:80/apex/soda/latest/MyCollection')]), more=False)
```

5. To insert a document, with content `{"name": "Alexander"}`, into the **MyCollection** collection, run this command:
curl -X POST -u user@example.com:password --data-binary @my.json -H "Content-Type: application/json" https://exadata1.oraclecorp.com/apex/soda/latest/MyCollection

Command result looks similar to this:

```json
{
  "items": [
    {
      "id": "2FFD968C531C49B9A7EAC4398DFFC02EE",
      "etag": "C1354F27A5180FF7B82BF01CBBC84022DCF5F7209DBF0E6DFCC626E3B0400C3",
      "lastModified": "2016-07-22T21:25:19.564394Z",
      "created": "2016-07-22T21:25:19.564394Z"
    }
  ],
  "hasMore": false,
  "count": 1
}
```

6. To retrieve the inserted document from the collection using the document unique ID, run this command:
Note:

In the following command example, the document ID shown is different from the actual document ID you must use to retrieve the document when you run the command. You must replace the document ID, 2FFD968C531C49B9A7EAC4398DFC02EE, with the auto-generated unique document ID you received after running the previous insert document command step.

curl -X GET -u user@example.com:password
https://exadata1.oraclecorp.com/apex/soda/latest/MyCollection/
2FFD968C531C49B9A7EAC4398DFC02EE

Command results:
The response body contains the document inserted in a previous step.

{"name" : "Alexander"}

7. To drop the collection, run this command:

curl -i -X DELETE -u user@example.com:password
https://exadata1.oraclecorp.com/apex/soda/latest/MyCollection

Command results contain:

... HTTP/1.1 200 OK ...

Use SODA for Java

SODA for Java is a Java library providing an implementation of SODA for use with Java. You can use it to perform create, remove, update, and delete (CRUD) operations on documents of any kind, and you can use it to query JSON documents.

For further details, see Oracle Database SODA for Java Developer's Guide.

Topics

- SODA for Java Pre-Requisites
- Get Started with SODA for Java

SODA for Java Pre-Requisites

Before you can use SODA for Java with Oracle Database Exadata Express Cloud Service, you need to configure your Java environment.

To use SODA for Java with Exadata Express:

- You must satisfy the software requirements for JDBC Thin Driver. See Software Requirements for JDBC Thin and UCP Connections.
- You must have the following JARs to use SODA for Java:
Get Started with SODA for Java

Follow these steps to compile and run a simple SODA for Java program that creates a JSON document collection, inserts documents into the collection, and retrieves documents using unique document keys or queries-by-example (QBEs).

For further details, see Database SODA for Java Developer's Guide.

To compile and run the sample program:

2. Configure JDBC Thin Driver for your service. See Connect with JDBC Thin Driver and UCP.
3. Enable SODA for your service. See Enable SODA for Your Service.
4. Ensure you have the appropriate JAR files. See SODA for Java Pre-Requisites.
5. Figure out the correct flags to use with the java command for your environment. See Property Settings for JDBC Thin Driver and UCP.

You are now ready to compile and run SODA Java programs.

6. Copy and paste the following simple SODA Java program into a file called testSODA.java. This program performs the following operations:
   - Create a new collection
   - Insert documents into the collection
   - Retrieve the first inserted document by its auto-generated key
   - Retrieve documents matching a query-by-example, or QBE

```java
import java.sql.Connection
import java.sql.DriverManager
import oracle.soda.rdbms.OracleRDBMSClient
import oracle.soda.OracleDatabase
import oracle.soda.OracleCursor
import oracle.soda.OracleCollection
import oracle.soda.OracleDocument
import oracle.soda.OracleException
import java.util.Properties
import oracle.jdbc.OracleConnection
```
public class testSODA{
    public static void main(String[] args) {

        // SODA works on top of a regular JDBC connection.
        // Set up the connection string: replace hostName, port, and
        // serviceName
        // with the info for your Oracle RDBMS instance
        String url = "jdbc:oracle:thin:user/password@service_name"

        OracleConnection conn = null
        try{
            // Get a JDBC connection to an Oracle instance
            conn = (OracleConnection) DriverManager.getConnection(url);

            // Enable JDBC implicit statement caching
            conn.setImplicitCachingEnabled(true);
            conn.setStatementCacheSize(50);

            // Get an OracleRDBMSClient - starting point of SODA for
            // Java application
            OracleRDBMSClient cl = new OracleRDBMSClient();

            // Get a database
            OracleDatabase db = cl.getDatabase(conn);
            // Create a collection with the name "MyFirstJSONCollection".
            // Note: Collection names are case-sensitive.
            // A table with the name "MyFirstJSONCollection" will be
            // created in the RDBMS to store the collection
            OracleCollection col =
                    db.admin().createCollection("MyFirstJSONCollection");

            // Create a few JSON documents, representing
            // users and the number of friends they have
            OracleDocument doc1 =
                    db.createDocumentFromString(
                            "{ "name" : "Alex", "friends" : "50" }" );

            OracleDocument doc2 =
                    db.createDocumentFromString(
                            "{ "name" : "Mia", "friends" : "300" }" );

            OracleDocument doc3 =
                    db.createDocumentFromString(
                            "{ "name" : "Gloria", "friends" : "399" }" );

            // Insert the documents into a collection, one-by-one.
            // The result documents contain auto-generated
            // keys, among other documents components (version, etc).
            // Note: SODA provides the more efficient bulk insert as well
            OracleDocument resultDoc1 = col.insertAndGet(doc1);
            OracleDocument resultDoc2 = col.insertAndGet(doc2);
            OracleDocument resultDoc3 = col.insertAndGet(doc3);

            // Retrieve the first document using its auto-generated
            // unique ID (aka key)
        }
    }
}
System.out.println("* Retrieving the first document by its key *
\n");
OracleDocument fetchedDoc =
col.find().key(resultDoc1.getKey()).getOne();
System.out.println(fetchedDoc.getContentAsString());

// Retrieve all documents representing users that have
// 300 or more friends. Use the following query-by-example:
// {friends : { $gte : 300}}.
System.out.println("\n* Retrieving documents representing users 
with" +
       " at least 300 friends \n");

OracleDocument f = db.createDocumentFromString("{ \"friends\" : { \"$gte\" : 300 }}");
OracleCursor c = null
try{
// Get a cursor over all documents in the collection
// that match our query-by-example
   c = col.find().filter(f).getCursor();

   while (c.hasNext()) {
      // Get the next document
      fetchedDoc = c.next();

      System.out.println(fetchedDoc.getContentAsString());
   }
}
finally{
   // Important: you must close the cursor to release resources!
   if (c != null) {
      c.close();
   }
}

// Drop the collection, deleting the table backing
// it and collection metadata
if (args.length > 0 && args[0].equals("drop")) {
   col.admin().drop();System.out.println("\n* Collection dropped *");
}
} catch (Exception e) {
   e.printStackTrace();
}
finally{
   if (conn != null) {
      try {
         conn.close();
      }
      catch (Exception e) {
      }
   }
}
7. Modify the url String at the beginning of the program with connection information for your service. Change the hostName:port and serviceName in the url String for your service:

For example, replace user, password and service_name in the following URL:

```java
String url = "jdbc:oracle:thin:user/password@service_name"
```

with:

- **user** - Replace user with the schema name.
- **password** - Replace password with the password for the schema. To use the default schema, see Find the Default Schema Name and Set the Password.
- **service_name** - Replace service_name with the service name listed in tnsnames.ora, which is downloaded as part of a prior step, see Download Client Credentials. For example, the content for tnsnames.ora looks something like this:

```java
cloud2 = (description=  
  (address= ...)  
  (connect_data= ...)  
  (security=...)  
}
```

In the above example, cloud2 is the service name.

8. Compile and run testSODA.java, making sure the necessary JAR files are in the classpath. For example, assuming you’re in the directory where the JARs are located, do:

```bash
javac "orajsoda-version.jar:ojdbc7.jar:javax.json-1.0.4.jar"
testSODA.java
java -Doracle.net.tns_admin=/home/user1/cloud  
-Djavax.net.ssl.trustStore=truststore.jks  
-Djavax.net.ssl.trustStorePassword=password  
-Djavax.net.ssl.keyStore=keystore.jks
```

**Note:**

The exact -D flags and their values shown in the following example will be different in your actual environment. The flags shown assume that /home/user1/cloud directory contains tnsnames.ora file, that truststore.jks and keystore.jks files are located in the current directory, that password is the trust store and key store password, and that JDK8 is used. See Property Settings for JDBC Thin Driver and UCP, for information on what the right flags are for your environment, and how to set them.
-Djavax.net.ssl.keyStorePassword=password
-Doracle.net.ssl_server_dn_match=true
-Doracle.net.ssl_version=1.2 "orajsoda-
version.jar:ojdbc7.jar:javax.json-1.0.4.jar:" testSODA

9. You should get the following output:

* Retrieving the first document by its key *

{ "name" : "Alex", "friends" : "50" }

* Retrieving documents representing users with at least 300 friends *

{ "name" : "Mia", "friends" : "300" }
{ "name" : "Gloria", "friends" : "399" }

Use SODA from SQLcl

SQLcl provides SODA support using commands and scripting integration.

Topics

- Get Started with SODA Using SQLcl Commands
- Get Started with SODA Using SQLcl Scripts

Get Started with SODA Using SQLcl Commands

Perform some basic SODA operations using SQLcl, including creating a collection, inserting documents, finding a select set of documents, committing the document insertions and dropping the collection.

Note:

For a complete list of all commands and additional information about each command, you can execute the soda help command within SQLcl.

To perform basic SODA operations using SQLcl.

1. Enable SODA for your service. See Enable SODA for Your Service.
2. Using Oracle SQLcl, connect to your Exadata Express service. See Connect SQLcl.
3. To create a collection named myColl, enter this command:

   SQL> soda create myColl

   Command results:

   Successfully created collection: myColl
4. To add a document with content \{"name" : "Alex"\} to myColl, enter this command:

   SQL> soda insert myColl \{"name" : "Alex"\}

   Command results:

   Json String inserted successfully

5. To add a document with content \{"name" : "Vlad"\} to myColl, enter this command:

   SQL> soda insert myColl \{"name" : "Vlad"\}

   Command results:

   Json String inserted successfully

6. To find all documents in myColl that have a "name" field that starts with "A", enter this command:

   SQL> soda find myColl -f \{"name" : { "$startsWith" : "A" }\}

   Command results:

   Key: BB325C9D97D7467FB34F20FEE3C3F091
   Content: \{"name" : "Alex"\}

   1 row selected.

   **Note:**

   Because the key value is automatically generated, the key value for your results will be different. This example shows how to run a query-by-example (QBE), also known as a filter (that's what the -f flag means). $startsWith is just one of the operators that can be used inside a QBE, the other ones are listed here: Filter Conditions in Oracle Database SODA for Java Developer's Guide. For comprehensive information on using QBEs, see Using Filter Specifications (QBEs) with SODA for Java in Oracle Database SODA for Java Developer's Guide.

7. To commit the document inserts, enter this command:

   SQL> commit;

   Command results:

   Commit complete.
8. To drop `myColl`, enter this command:

```
SQL> soda drop myColl
```

Command results:

```
Successfully dropped: myColl
```

Get Started with SODA Using SQLcl Scripts

You can access SODA for Java classes using SQLcl running Javascript. The following simple script illustrates using SODA for Java classes with Javascript in SQLcl. The script creates a collection, inserts a document, and finds the inserted document using a QBE (query-by-example).

For further details about Java classes, see SODA for Java Javadoc and Oracle Database SODA for Java Online Documentation.

To run a simple Javascript that performs some basic SODA operations:

1. Enable SODA for your service. See Enable SODA for Your Service.
2. Using Oracle SQLcl, connect to your Exadata Express service. See Connect SQLcl.
3. Copy the following script into a file named `soda.sql`.

```javascript
script
ctx.write('Caffeinated SODA \n');

var OracleRDBMSClient = Java.type("oracle.soda.rdbms.OracleRDBMSClient");
var OracleDataSource = Java.type('oracle.jdbc.pool.OracleDataSource');
var cl = new OracleRDBMSClient();
var db = cl.getDatabase(conn);

ctx.write("Creating foo\n")
foo = db.admin().createCollection("foo");

var doc = db.createDocumentFromString("{ "name" : "Alex", "friends
" : "50" }"));
ctx.write("Inserted Document\n")
foo.insertAndGet(doc);

conn.commit();

var f = db.createDocumentFromString(JSON.stringify( {"name" : ("$startsWith" : "A" ))});
```
ctx.write("Searching...
")
var c = foo.find().filter(f).getCursor();

while (c.hasNext()) {
    cDoc = c.next();
    ctx.write(cDoc.getKey() + "\n");
    ctx.write(cDoc.getContentAsString());
    ctx.write("\n\n");
}
ctx.write("\n\n");

4. To execute the script file, enter this command from within SQLcl:

SQL> @soda.sql

Command results:

Caffeinated SODA
Creating foo
Inserted Document
Searching...8AC72C7E5F2243B99F56E1362B80DF689
{"name" : "Alex", "friends" : "50" }

Note:
If you run this script multiple times, the collection grows. Each time the script is executed, another JSON document is inserted. This is because createCollection(...) simply opens the collection if it already exists. To completely drop the collection created by the script, execute the soda drop foo command from SQLcl.
Develop on the Database

The Oracle Database Exadata Express Cloud Service allows you to connect your PDB from diverse database clients over Oracle Net Services (SQL*Net). After connecting to a database client, you can develop on your database in the Exadata Express leveraging the features supported by each database client.

Topics
- Develop with Oracle SQL Developer
- Use Oracle Application Express Application Data Load Utility
- Develop with Oracle SQLcl
- Develop with Oracle SQL*Plus
- Manage Data with Oracle Application Express SQL Workshop

Develop with Oracle SQL Developer

SQL Developer provides a powerful integrated development environment (IDE) for authoring SQL and PL/SQL on databases. You can use SQL Developer to connect to your Oracle Database Exadata Express Cloud Service and develop on the database. You can use Oracle SQL Developer to load data to Exadata Express service from either an on-premise database or other cloud service.

Oracle SQL Developer provides various options for loading data into your database. Although you can use any of these options, each of them will be more beneficial in a specific scenario, as explained in individual topics. Depending on your requirement, you can choose the most suitable option to perform data loading.

Topics
- Use Database Copy for Data Load
- Use SQL Developer Cart for Data Load
- Use Drag and Drop for Data Load
- Execute SQL Scripts for Data Loading

To learn more about SQL Developer, see About SQL Developer in the SQL Developer User’s Guide.

Use Database Copy for Data Load

Oracle SQL Developer provides an option called Database Copy to choose and copy database objects into your Oracle Database Exadata Express Cloud Service from an on-premise connection or any other cloud connection. This option is more suitable...
when you want to copy objects up and have the ability to filter data out using WHERE clauses or pull all or selected objects from one or more schemas.

To copy database objects from another connection into your Exadata Express, you must perform the following steps:

1. Install Oracle SQL Developer locally, and create a cloud connection from Oracle SQL Developer to your service. see Connect SQL Developer.

2. From Oracle SQL Developer, click Tools.
   The Tools drop down menu displays.

3. From the drop down menu, click Database Copy.
   The Database Copy Wizard appears.
4. Follow the on-screen instructions, and select the suitable options as you proceed through the various steps of the wizard.

5. **Click Finish.**

The Copying dialog displays.

All the database objects specified in the Database Copy Wizard are copied to your Exadata Express connection in Oracle SQL Developer. You can browse the Connections tree to verify if the selected database objects are copied in your Exadata Express connection.

**Use SQL Developer Cart for Data Load**

Oracle SQL Developer creates a cart containing objects you want to load in to your Oracle Database Exadata Express Cloud Service, connects to your service and deploys data from the cart to the service. Using cart is more convenient, when you need to create a subset of objects from one or more schemas locally that you want to deploy to another database on a regular basis.

To deploy objects to your service using cart, you must perform the following steps:
1. Install Oracle SQL Developer locally, and create a cloud connection from Oracle SQL Developer to your service. See Connect SQL Developer.

2. From Oracle SQL Developer, click View. The View drop down menu displays.

3. From the drop down menu, click Cart. The Cart window appears on the bottom right.

4. Drag the required Oracle Database objects from your on-premise database, and drop them in to the Cart window on the right.
5. If you want to include data with the cart deployment, in the Cart window, click the **Data** check box.

6. To deploy this cart, click the **Copy (Ctrl+C)** icon at the top right of the Cart window.

The Copy Objects dialog displays.

7. Select your Exadata Express connection for Destination Connection. Choose the appropriate options under Copy DDL and Copy Data categories.

8. Click **Apply**.

The Copying dialog displays.
All the database objects added to the cart are copied to your Exadata Express connection in Oracle SQL Developer. You can browse the Connections tree to verify if the selected database objects are copied in your Exadata Express connection.

Use Drag and Drop for Data Load

Oracle SQL Developer provides an option to drag and drop database objects into your Oracle Database Exadata Express Cloud Service from an on-premise connection or any other cloud connection. Drag and Drop option is perfect for ad-hoc, on-the-fly operations. For example, you can just select a table or view in the connections tree, and drag it over when you need to set up a table or view quickly.

To drag and drop a database object from another connection into your Exadata Express, you must perform the following steps:

1. Install Oracle SQL Developer locally, and create a cloud connection from Oracle SQL Developer to your service. see Connect SQL Developer.

2. From Oracle SQL Developer, select one or more database objects in the source connection. Drag it to the destination location in your Exadata Express service in the connections tree, and drop it.

   The Copy To Oracle dialog opens. It provides you with options to choose to copy DDL or data or both. Under each category, you have options to choose an action to be performed if the database object already exists in your Exadata Express service.
Note:

- The source database must be connected before performing the Drag and Drop operation. If you drop on a disconnected database, Oracle SQL Developer will open/connect it, for you to copy.
- You can also drag an entire node, such as 'tables' and it will drag all the tables, and their related objects such as triggers, indexes, and constraints, for dropping into your destination database.

3. Select the required options, and click **OK**.

The Copying dialog displays.
4. Follow the on-screen instructions, and the selected database object will be copied into your Exadata Express connection in Oracle SQL Developer.

You can browse the Connections tree to verify if the selected database objects are copied in your Exadata Express connection. You can also look at the Database Copy report which is provided at the end of the copy activity.

**Execute SQL Scripts for Data Loading**

You can execute SQL*Plus scripts from Oracle SQL Developer to load data into your Oracle Database Exadata Express Cloud Service. This option is more beneficial for the users that already have deployment scripts. If required, they can even open these scripts and modify them using the powerful SQL Worksheet with error highlighting, SQL Recall, Code Insight, formatting and so on, before executing them.

Follow these steps to load and execute SQL scripts from Oracle SQL Developer:

1. Install Oracle SQL Developer locally, and create a cloud connection from Oracle SQL Developer to your service. see Connect SQL Developer.

2. Click Files from the View menu.

   The Files explorer tab appears.

3. Browse to the folder where the SQL script resides, and double-click the script to be executed.

   **Note:**

   The file having the SQL script must be with a .sql extension.

4. The SQL script loads into the SQL Worksheet. Click the Run Script (F5) icon.

   The script executes, and loads the data into the required tables as per the script.

**Use Oracle Application Express Application Data Load Utility**

Applications with data loading capability allow application end users to dynamically import data into a table within any schema to which the user has access. End users run a Data Load Wizard that uploads data from a file or copies and pastes data entered by the end user directly into the wizard.

To learn more about developing Oracle Application Express applications with this capability, see Creating a Data Load Wizard in Oracle Application Express Application Builder User's Guide.

To learn more about how an application end user uses this capability, see Data Loading Wizard Examples in Oracle Application Express End User Guide.
Develop with Oracle SQLcl

Oracle SQLcl is a powerful java-based command line interface (CLI) for Oracle Database. You can use SQLcl to connect to your Oracle Database Exadata Express Cloud Service and develop on the database.

Topics:
- Manage Database Objects with SQLcl
- Query and Manipulate Data Using SQLcl
- Use the help command in SQLcl
- Use SQLcl to Data Load

Manage Database Objects with SQLcl

Oracle SQLcl allows you to create and manage various database objects in your Oracle Database Exadata Express Cloud Service.

To execute basic commands for managing the database objects using SQLcl:

1. Using SQLcl, connect to your Exadata Express service. See Connect SQLcl.

2. To create a table named emp, enter this command:

   ```sql
   SQL> CREATE TABLE emp(
      2     empno NUMBER(5) PRIMARY KEY,
      3     ename VARCHAR2(15) NOT NULL,
      4     job VARCHAR2(10),
      5     mgr NUMBER(5),
      6     hiredate DATE DEFAULT(sysdate),
      7     deptno NUMBER(3) NOT NULL);
   
   Command results:
   
   Table EMP created.
   
   3. To add a new column named email to the emp table, enter this command:

   ```sql
   SQL> ALTER TABLE emp
         2   ADD email VARCHAR2(15);
   
   Command results:
   
   Table EMP altered.
   
   4. To rename a column named email to email_address in the emp table, enter this command:

   ```sql
   SQL> ALTER TABLE emp RENAME COLUMN email TO email_address;
   ```
5. To delete a column named email_address from the emp table, enter this command:

   SQL> ALTER TABLE emp DROP COLUMN email_address;

Command results:

Table EMP altered.

6. To rename the emp table as temp, enter this command:

   SQL> RENAME emp TO temp;

Command results:

Table renamed.

7. To drop the temp table, enter this command:

   SQL> DROP TABLE temp;

Command results:

Table TEMP dropped.

To know more about the SQL commands used to manage the database objects, see Managing Tables in Database Administrator’s Guide.

Query and Manipulate Data Using SQLcl

Oracle SQLcl allows you to query and manipulate data in your Oracle Database Exadata Express Cloud Service by entering commands in the command-line.

To execute basic commands for querying and manipulating data using SQLcl:

1. Using SQLcl, connect to your Exadata Express service. See Connect SQLcl.

2. To retrieve all the records of employees whose salary is greater than 12000, from the table named employees, enter this command:

   SQL> select * from employees where salary > 12000;

The records for all the employees whose salary is greater than 12000 are displayed without any formatting.

<table>
<thead>
<tr>
<th>EMPLOYEE_ID</th>
<th>FIRST_NAME</th>
<th>LAST_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMAIL</td>
<td>PHONE_NUMBER</td>
<td>HIRE_DATE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EMPLOYEE_ID</th>
<th>FIRST_NAME</th>
<th>LAST_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMAIL</td>
<td>PHONE_NUMBER</td>
<td>HIRE_DATE</td>
</tr>
<tr>
<td>100</td>
<td>Steven</td>
<td>King</td>
</tr>
<tr>
<td>101</td>
<td>Neena</td>
<td>Kochhar</td>
</tr>
<tr>
<td>102</td>
<td>Lex</td>
<td>De Haan</td>
</tr>
<tr>
<td>108</td>
<td>Nancy</td>
<td>Greenberg</td>
</tr>
<tr>
<td>145</td>
<td>John</td>
<td>Russell</td>
</tr>
<tr>
<td>146</td>
<td>Karen</td>
<td>Partners</td>
</tr>
</tbody>
</table>
### 3. Displaying records in formatted way using SQLcl

As you can see that the above query's result is not formatted and hence difficult to associate the values with their corresponding columns. Using SQLcl, you can display the records from a select query in a formatted way. The `SET SQLFORMAT` command helps you format your query results.

```sql
SQL> set sqlformat csv
SQL> select * from employees where salary > 12000;
```

This time, the records from the select query are displayed in csv format:

```
"EMPLOYEE_ID","FIRST_NAME","LAST_NAME","EMAIL","PHONE_NUMBER","HIRE_DATE","JOB_ID","SALARY","COMMISSION_PCT","MANAGER_ID","DEPARTMENT_ID"
100,"Steven","King","SKING","515.123.4567",17-JUN-03,"AD_PRES",24000,,90
101,"Neena","Kochhar","NKOCHHAR","515.123.4568",21-SEP-05,"AD_VP",17000,,100,90
102,"Lex","De Haan","LDEHAAN","515.123.4569",13-JAN-01,"AD_VP",17000,,100,90
145,"John","Russell","JRUSSEL","011.44.1344.429268",01-OCT-04,"SA_MAN",14000,0.4,100,80
146,"Karen","Partners","KPARTNER","011.44.1344.467268",05-
```
JAN-05, "SA_MAN", 13500, 0.3, 100, 80

8 rows selected.

4. To display the output of a query in ansiconsole format, enter this command:

SQL> set sqlformat ansiconsole
SQL> select * from employees where salary > 12000;

The records from the select query are displayed in ansiconsole format. The ansiconsole formats the data to best fit the display for each page of results. SQL*Plus style report formatting like COL will be ignored in this mode.

EMPLOYEE_ID  FIRST_NAME  LAST_NAME  EMAIL     PHONE_NUMBER  
HIRE_DATE  JOB_ID   SALARY  COMMISSION_PCT
MANAGER_ID  DEPARTMENT_ID
100          Steven      King       SKING     515.123.4567        17-JUN-03  AD_PRES  24000
101          Neena       Kochhar    NKOCHHAR  515.123.4568        21-SEP-05  AD_VP    17000
102          Lex         De Haan    LDEHAAN   515.123.4569        13-JAN-01  AD_VP    17000
108          Nancy       Greenberg  NGREENBE  515.124.4569        17-AUG-02  FI_MGR   12008
145          John        Russell    JRUSSEL   011.44.1344.429268  01-OCT-04  SA_MAN   14000
146          Karen       Partners   KPARTNER  011.44.1344.467268  05-JAN-05  SA_MAN   13500
201          Michael     Hartstein  MHARTSTE  515.123.5555        17-FEB-04  MK_MAN   13000
205          Shelley     Higgins    SHIGGINS  515.123.8080        07-JUN-02  AC_MGR   12008

8 rows selected.
The `set sqlformat <setting>` command allows you to display the output in many formats like html, json, insert, xml, loader, or fixed. Executing `set sqlformat default`, will return the formatting to default. To know more about `set sqlformat <setting>` command, use the `HELP <command>`. See Use the `help` command in SQLcl.

5. To insert a new record for a country named Hongkong in the countries table, enter this command:

   SQL> INSERT INTO countries(country_id,country_name,region_id) VALUES('HK','HONGKONG',3);

   Command results:
   1 row inserted.

6. To modify the value of country_code for the record inserted above, enter this command:

   SQL> UPDATE countries SET country_id = 'HN' WHERE country_id = 'HK';

   Command results:
   1 row updated.

7. To delete the record created in the step above, from countries table, enter this command:

   SQL> DELETE FROM countries WHERE country_id = 'HN';

   Command results:
   1 row deleted.

8. To see the list of columns in the employees table, enter this command:

   SQL> describe employees;

   Command results:

<table>
<thead>
<tr>
<th>Name</th>
<th>Null?</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE_ID</td>
<td>NOT NULL</td>
<td>NUMBER(6)</td>
</tr>
<tr>
<td>FIRST_NAME</td>
<td></td>
<td>VARCHAR2(20)</td>
</tr>
<tr>
<td>LAST_NAME</td>
<td>NOT NULL</td>
<td>VARCHAR2(25)</td>
</tr>
<tr>
<td>EMAIL</td>
<td>NOT NULL</td>
<td>VARCHAR2(25)</td>
</tr>
<tr>
<td>PHONE_NUMBER</td>
<td>NOT NULL</td>
<td>VARCHAR2(20)</td>
</tr>
<tr>
<td>HIRE_DATE</td>
<td>NOT NULL</td>
<td>DATE</td>
</tr>
</tbody>
</table>
9. The DESC command is a SQL*plus command that SQLcl supports. Additionally, SQLcl supports `INFO[s] <schema.object>` which provide more details about the objects requested as, column comments, indexes, references, and so on.

```
SQL> information employees
```

Command results:

```
TABLE: EMPLOYEES
  LAST ANALYZED:2016-01-12 22:00:38.0
  ROWS :107
  SAMPLE SIZE :107
  INMEMORY :DISABLED

Columns

<table>
<thead>
<tr>
<th>NAME</th>
<th>DATA TYPE</th>
<th>NULL</th>
<th>DEFAULT</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>*EMPLOYEE_ID</td>
<td>NUMBER(6,0)</td>
<td>No</td>
<td></td>
<td>Primary key of employees table.</td>
</tr>
<tr>
<td>FIRST_NAME</td>
<td>VARCHAR2(20 BYTE)</td>
<td>Yes</td>
<td></td>
<td>First name of the employee. A not null column.</td>
</tr>
<tr>
<td>LAST_NAME</td>
<td>VARCHAR2(25 BYTE)</td>
<td>No</td>
<td></td>
<td>Last name of the employee. A not null column.</td>
</tr>
<tr>
<td>EMAIL</td>
<td>VARCHAR2(25 BYTE)</td>
<td>No</td>
<td></td>
<td>Email id of the employee</td>
</tr>
<tr>
<td>PHONE_NUMBER</td>
<td>VARCHAR2(20 BYTE)</td>
<td>Yes</td>
<td></td>
<td>Phone number of the employee; includes country code and area code</td>
</tr>
<tr>
<td>HIRE_DATE</td>
<td>DATE</td>
<td>No</td>
<td></td>
<td>Date when the employee started on this job. A not null column.</td>
</tr>
<tr>
<td>JOB_ID</td>
<td>VARCHAR2(10 BYTE)</td>
<td>No</td>
<td></td>
<td>Current job of the employee; foreign key to job_id column of the jobs table. A not null column.</td>
</tr>
<tr>
<td>SALARY</td>
<td>NUMBER(8,2)</td>
<td>Yes</td>
<td></td>
<td>Monthly salary of the employee. Must be greater than zero (enforced by constraint emp_salary_min)</td>
</tr>
<tr>
<td>COMMISSION_PCT</td>
<td>NUMBER(2,2)</td>
<td>Yes</td>
<td></td>
<td>Commission percentage of the employee; Only employees in sales</td>
</tr>
</tbody>
</table>
```
department eligible for commission percentage

MANAGER_ID NUMBER(6,0) Yes Manager id of the employee; has same domain as manager_id in departments table. Foreign key to employee_id column of employees table.

(Useful for reflexive joins and CONNECT BY query)

DEPARTMENT_ID NUMBER(4,0) Yes Department id where employee works; foreign key to department_id column of the departments table.

Indexes

<table>
<thead>
<tr>
<th>INDEX_NAME</th>
<th>UNIQUENESS</th>
<th>STATUS</th>
<th>FUNCIDX_STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR.EMP_JOB_IX</td>
<td>NONUNIQUE</td>
<td>VALID</td>
<td>JOB_ID</td>
</tr>
<tr>
<td>HR.EMP_NAME_IX</td>
<td>NONUNIQUE</td>
<td>VALID</td>
<td>LAST_NAME,</td>
</tr>
<tr>
<td>HR.EMP_EMAIL_UK</td>
<td>UNIQUE</td>
<td>VALID</td>
<td>EMAIL</td>
</tr>
<tr>
<td>HR.EMP_EMP_ID_PK</td>
<td>UNIQUE</td>
<td>VALID</td>
<td>EMPLOYEE_ID</td>
</tr>
<tr>
<td>HR.EMP_MANAGER_IX</td>
<td>NONUNIQUE</td>
<td>VALID</td>
<td>MANAGER_ID</td>
</tr>
<tr>
<td>HR.EMP_DEPARTMENT_IX</td>
<td>NONUNIQUE</td>
<td>VALID</td>
<td>DEPARTMENT_ID</td>
</tr>
</tbody>
</table>

References

<table>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>CONSTRAINT_NAME</th>
<th>DELETE_RULE</th>
<th>STATUS</th>
<th>DEFERRABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPARTMENTS</td>
<td>DEPT_MGR_FK</td>
<td>NO ACTION</td>
<td>ENABLED</td>
<td>NOT DEFERRABLE</td>
</tr>
<tr>
<td>EMPLOYEES</td>
<td>EMP_MANAGER_FK</td>
<td>NO ACTION</td>
<td>ENABLED</td>
<td>NOT DEFERRABLE</td>
</tr>
<tr>
<td>JOB_HISTORY</td>
<td>JHIST_EMP_FK</td>
<td>NO ACTION</td>
<td>ENABLED</td>
<td>NOT DEFERRABLE</td>
</tr>
</tbody>
</table>

Note:

Use the SQLcl Help <command> to know about other SQLcl commands. See Use the help command in SQLcl.

To know more about the SQL commands used to manage the data, see Querying and Manipulating Data in Database 2 Day Developer’s Guide.

Use the help command in SQLcl

Oracle SQLcl provides help command to get assistance in using any command supported by SQLcl.
Typing `help` from the SQLcl command line, after connecting to your Oracle Database Exadata Express Cloud Service, lists all the commands supported by SQLcl.

```
SQL> help
For help on a topic type help<topic>
List of Help topics available:
/
@@
ACCEPT
ALIAS
APEX
APPEND
ARCHIVE LOG
BREAK
BRIDGE
BTITLE
CD
CHANGE
CLEAR
COLUMN
COMPUTE
CONNECT
COPY
CTAS
DDL
DEFINE
DEL
DESCRIBE
DISCONNECT
EDIT
EXECUTE
EXIT
FORMAT
GET
HISTORY
HOST
INFORMATION
INPUT
LIST
LOAD
NET
NOHISTORY
OERR
PASSWORD
PAUSE
PRINT
PROMPT
QUIT
REMARK
REPEAT
RESERVED WORDS
REST
RUN
SAVE
```
By executing `help <command>` will print the help documentation for that command.

```
SQL> help information
INFORMATION
--------

This command is like describe but with more details about the objects requested.

INFO[RMATION] {[schema.]object[@connect_identifier]}
INFO+ will show column statistics
```

**Use SQLcl to Data Load**

Oracle SQLcl, allows you to load data to your Oracle Database Exadata Express Cloud Service using multiple insert statements or from a csv file.

**Pre-requisites**

In order to load data into your Exadata Express service from an external data file, ensure that:

- The first row of the file is a header row, and the columns in the header row match the columns defined on the table.
- The columns are delimited by a comma and optionally enclosed in double quotes.
- The lines are terminated with standard line terminators for windows, unix or mac.
- The file is encoded UTF8.
Procedure

Follow these steps, to perform data loading using SQLcl:

1. Using SQLcl, connect to your Exadata Express service. See Connect SQLcl.
2. At the SQLcl command prompt, type `load [<schema>.table_name> <.csv file name>`.

   All the records listed in the csv file are loaded into the table.

**Note:**

- The load is processed with 50 rows per batch.
- If `AUTOCOMMIT` is set in SQLCL, a commit is done every 10 batches.
- The load is terminated if more than 50 errors are found.

Develop with Oracle SQL*Plus

You can use SQL*Plus to connect to your Oracle Database Exadata Express Cloud Service and develop on the database.

Topics:

- Enter and Execute Commands
- Manage Database Objects with SQL*Plus
- Query and Manipulate Data Using SQL*Plus
- Use SQL*Plus to Data Load

Enter and Execute Commands

After you connect to your Oracle Database Exadata Express Cloud Service using SQL*Plus, you can execute commands from the command-line to develop the database.

To execute a command from SQL*Plus, type the command in the command-line, and press the Return key. The SQL*Plus allows you to enter three types of commands as:

- SQL commands
- PL/SQL blocks
- SQL*Plus commands

To learn more about the types of commands that can be executed, see Entering and Executing Commands in SQL*Plus® User's Guide and Reference.
Manage Database Objects with SQL*Plus

Oracle SQL*Plus allows you to manage various database objects in your Oracle Database Exadata Express Cloud Service by entering commands in the command-line.

To execute basic commands for managing the database objects using SQL*Plus:

1. Using SQL*Plus, connect to your Exadata Express service. See Connect SQL*Plus.

2. To create a table named emp, enter this command:

   SQL> CREATE TABLE emp(
   2  empno NUMBER(5) PRIMARY KEY,
   3  ename VARCHAR2(15) NOT NULL,
   4  job VARCHAR2(10),
   5  mgr NUMBER(5),
   6  hiredate DATE DEFAULT(sysdate),
   7  deptno NUMBER(3) NOT NULL);

   Command results:

   Table created.

3. To add a new column named email to the emp table, enter this command:

   SQL> ALTER TABLE emp
   2  ADD email VARCHAR2(15);

   Command results:

   Table altered.

4. To rename a column named email to email_address in the emp table, enter this command:

   SQL> ALTER TABLE emp RENAME COLUMN email TO email_address;

   Command results:

   Table altered.

5. To delete a column named email_address from the emp table, enter this command:

   SQL> ALTER TABLE emp DROP COLUMN email_address;
Command results:

Table altered.

6. To rename the emp table as temp, enter this command:

   SQL> RENAME emp TO temp;

Command results:

Table renamed.

7. To drop the temp table, enter this command:

   SQL> DROP TABLE temp;

Command results:

Table dropped.

To know more about the SQL commands used to manage the database objects, see Managing Tables in Database Administrator's Guide.

Query and Manipulate Data Using SQL*Plus

Oracle SQL*Plus allows you to query and manipulate data in your Oracle Database Exadata Express Cloud Service by entering commands in the command-line.

To execute basic commands for querying and manipulating data using SQL*Plus:

1. Using SQL*Plus, connect to your Exadata Express service. See Connect SQL*Plus.

2. To retrieve all the records from the table named countries, enter this command:

   SQL> select * from countries;

Command results:

<table>
<thead>
<tr>
<th>CO</th>
<th>COUNTRY_NAME</th>
<th>REGION_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>Argentina</td>
<td>2</td>
</tr>
<tr>
<td>AU</td>
<td>Australia</td>
<td>3</td>
</tr>
<tr>
<td>BE</td>
<td>Belgium</td>
<td>1</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
<td>2</td>
</tr>
<tr>
<td>CA</td>
<td>Canada</td>
<td>2</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
<td>1</td>
</tr>
<tr>
<td>CN</td>
<td>China</td>
<td>3</td>
</tr>
<tr>
<td>DE</td>
<td>Germany</td>
<td>1</td>
</tr>
<tr>
<td>DK</td>
<td>Denmark</td>
<td>1</td>
</tr>
<tr>
<td>EG</td>
<td>Egypt</td>
<td>4</td>
</tr>
<tr>
<td>FR</td>
<td>France</td>
<td>1</td>
</tr>
</tbody>
</table>
3. To insert a new record for a country named Hongkong in the countries table, enter this command:

```sql
SQL> INSERT INTO countries(country_id,country_name,region_id)
VALUES('HK','HONGKONG',3);
```

Command results:

1 row created.

4. To modify the value of country_code for the record inserted above, enter this command:

```sql
SQL> UPDATE countries SET country_id = 'HN' WHERE country_id = 'HK';
```

Command results:

1 row updated.

5. To delete the record created in the step above, from countries table, enter this command:

```sql
SQL> DELETE FROM countries WHERE country_id = 'HN';
```

Command results:

1 row deleted.
To see the list of columns in the countries table, enter this command:

```
SQL> DESC countries;
```

Command results:

<table>
<thead>
<tr>
<th>Name</th>
<th>Null?</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNTRY_ID</td>
<td>NOT NULL</td>
<td>CHAR(2)</td>
</tr>
<tr>
<td>COUNTRY_NAME</td>
<td>NOT NULL</td>
<td>VARCHAR2(40)</td>
</tr>
<tr>
<td>REGION_ID</td>
<td></td>
<td>NUMBER</td>
</tr>
</tbody>
</table>

To know more about the SQL commands used to manage the data, see Querying and Manipulating Data in Database 2 Day Developer's Guide.

Use SQL*Plus to Data Load

Using Oracle SQL*Plus, you can load data from an on-premise database to your Oracle Database Exadata Express Cloud Service or vice-versa. Similarly, you can also use the SQL*Plus features to load data between your Exadata Express service and any other cloud service which supports connecting through SQL*Plus.

SQL*Plus supports loading bulk data or copying database objects from one database to another by various means. As your Exadata Express service allows you to connect from SQL*Plus, it also supports features to load data between your database on the service and any other database either on cloud or on-premise. The following options are available for you to load data to/from your Exadata Express service:

**Running SQL scripts to create database objects & load data.**

Follow these steps, to execute SQL scripts with Exadata Express service:

1. Using SQL*Plus, connect to your Exadata Express service. See Connect SQL*Plus.

2. To execute a script of SQL commands to load data into your Exadata Express service, execute the following commands at the command prompt:

```
START <filename> or @<filename>
```

SQL*Plus runs the commands in the file, and displays the status of the execution.

**Note:**

SQL*Plus assumes the file has a .sql extension by default.

For further information on using SQL*Plus for executing SQL scripts, see Running Scripts in SQL*Plus® User's Guide and Reference.
Using SQL *Loader Utility

A typical SQL*Loader session takes as input a control file, which controls the behavior of SQL*Loader, and one or more datafiles. The output of SQL*Loader is an Oracle database (where the data is loaded), a log file, a bad file, and potentially, a discard file. It has a powerful data parsing engine that puts little limitation on the format of the data in the datafile. SQL Loader conventional path and direct path work remotely as long as there is a service for the target database that the user can provide in the connect string. It can be run from a remote machine where the data files are present, and therefore, there is no need to send files to the target machine using SFTP.

For further information, see SQL*Loader Command-Line Reference in Oracle Database Utilities.

Manage Data with Oracle Application Express SQL Workshop

The Oracle Application Express SQL Workshop provides you with tools to manage data in your Oracle Database Exadata Express Cloud Service.

Topics:

- Use Oracle Application Express SQL Workshop to Develop the Database
- Use Oracle Application Express SQL Workshop Data Upload Utility
- Use Oracle Application Express SQL Workshop to Export Data
- Upload Data with SQL Scripts

Use Oracle Application Express SQL Workshop to Develop the Database

The Oracle Application Express SQL Workshop Object Browser enables developers to browse, create, and edit objects in a database. Oracle Application Express has many views defined to help you work with the underlying table structures. The views combine multiple base tables and use meaningful column names to minimize complexity and to be more intuitive to use.

See Oracle Application Express API Reference for details of the views defined.

To manage data using Object Browser:

1. Access Oracle Application Express Application Builder from Oracle Database Exadata Express Cloud Service service console. See Access Oracle Application Express.
   The Oracle Application Express components displays.

2. Click Go to SQL Workshop.
   The Oracle Application Express SQL Workshop home page displays.

3. Click Object Browser.
4. Browse, create, edit and delete database objects.
   To learn more, see Managing Database Objects with Object Browser in Oracle Application Express SQL Workshop Guide.

Use Oracle Application Express SQL Workshop Data Upload Utility

The Data Upload utility enables you to upload data from a text file, an XML document or a spreadsheet. The utility is accessed from the Oracle Application Express SQL Workshop.

Note:
Files uploaded with Oracle Application Express SQL Workshop must not exceed 100MB. For files larger than 100MB, please use Oracle SQL Developer.

To upload data using the Data Upload utility:

1. Access Oracle Application Express Application Builder from Oracle Database Exadata Express Cloud Service service console. See Access Oracle Application Express.
   The Oracle Application Express components displays.

2. Click Go to SQL Workshop.
   The Oracle Application Express SQL Workshop home page displays.

3. Click Utilities.

4. Click Data Workshop.

5. Under the Data Load category, click one of the following types of data:
   • Text Data
   • XML Data
   • Spreadsheet Data. See Use Oracle Application Express Data Upload with Spreadsheet

6. Follow on-screen instructions.

To learn more about uploading data with the Data Upload utility, see Loading Data in Oracle Application Express SQL Workshop Guide.

Use Oracle Application Express Data Upload with Spreadsheet

The Oracle Application Express Data Upload utility enables you to upload data from a text file, an XML document or a spreadsheet. The utility is accessed from the Oracle Application Express SQL Workshop.

1. Access Oracle Application Express Application Builder from Oracle Database Exadata Express Cloud Service service console. See Access Oracle Application Express.
The Oracle Application Express components displays.

2. Click **Go to SQL Workshop**.
   The Oracle Application Express SQL Workshop home page displays.

3. Click **Utilities**.

4. Click **Data Workshop**.

5. Under Data Load category, click **Spreadsheet Data**.
   The Load Data - Target and Method wizard appears.

6. Select a suitable option for **Load To:** and **Upload file (comma separated or tab delimited)** for **Load From:**.

   ![Load Data - Target and Method](image)

   **Note:**
   You can load the data from a csv file, or even copy and paste the csv formatted data directly in the copy-paste area in this wizard. To copy the data directly into the wizard, select **Copy and Paste** for **Load From:**

7. Click **Next >**.

8. Choose the file for data upload, select the values for other fields on the wizard as required, and click **Next >**.

   **Note:**
   The file to be uploaded must be text-based. To upload a .XLS file, first save it as CSV.

To learn more about uploading data with the Data Upload utility, see Using Data Workshop in *Oracle Application Express SQL Workshop Guide*.

**Use Oracle Application Express SQL Workshop to Export Data**

The Data Unload utility provided by Oracle Application Express SQL Workshop, enables you to export data from your Oracle Database Exadata Express Cloud Service into a text file, an XML document or a spreadsheet.

**To export data using the Data Unload utility:**

1. Access Oracle Application Express Application Builder from Oracle Database Exadata Express Cloud Service service console. See *Access Oracle Application Express*.
   
   The Oracle Application Express components displays.

2. Click **Go to SQL Workshop**.
   
   The Oracle Application Express SQL Workshop home page displays.

3. Click **Utilities**.

4. Click **Data Workshop**.

5. Under the Data Unload category, click one of the following options depending on the required format for the exported data:
   
   - to Text
   - to XML

6. Follow on-screen instructions.

To learn more about exporting data with the Data Unload utility, see Unloading Data in *Oracle Application Express SQL Workshop Guide*.

**Upload Data with SQL Scripts**

The SQL Scripts functionality allows you to create, upload, and run SQL scripts that insert data into the database. You could use an SQL script to load data into your Oracle Database Exadata Express Cloud Service.

**Note:**

Files uploaded with Oracle Application Express SQL Workshop must not exceed 100MB. For files larger than 100MB, please use Oracle SQL Developer.

These operations are accessed from the Oracle Application Express SQL Workshop.

**To insert data using SQL Scripts:**

1. Invoke Oracle Application Express components from the Oracle Database Exadata Express Cloud Service service console. See *Access Oracle Application Express*.
   
   The Oracle Application Express components displays.
2. Click **Go to SQL Workshop**.
   The Oracle Application Express SQL Workshop home page displays.

3. Click **SQL Scripts**.

4. Create or upload an SQL script.
   To learn more, see Creating a SQL Script in the Oracle Application Express SQL Workshop Guide.

5. Click the **Run** icon for the SQL script you want to execute.
   To learn more about running SQL scripts, see Executing a SQL Script in Oracle Application Express SQL Workshop Guide.

   To learn more about using SQL scripts to load data, see Using SQL Scripts in Oracle Application Express SQL Workshop Guide.
Manage Web Services

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 protocol over HTTP. 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. Your Oracle Database Exadata Express Cloud Service allows implementing and calling both SOAP and RESTful style web services.

Topics

• Call SOAP and RESTful Web Services
• Implement RESTful Web Services with Oracle SQL Developer
• Implement RESTful Web Services with Oracle Application Express
• Consume RESTful Web Services with JET
• Troubleshoot RESTful Web Services

Call SOAP and RESTful Web Services

The APEX_WEB_SERVICE package enables you to integrate other systems with Oracle Application Express by allowing you to interact with Web services anywhere you can use PL/SQL in your application. The API contains procedures and functions to call both SOAP and RESTful style Web services.

It contains functions to parse the responses from Web services and to encode/decode in to SOAP friendly base64 encoding. This API also contains package globals for managing cookies and HTTP headers when calling Web services whether from the API or by using standard processes of type Web service. Cookies and HTTP headers can be set before invoking a call to a Web service by populating the globals and the cookies and HTTP headers returned from the Web service response can be read from other globals.
**Note:**

All Web services must be secured. Only HTTPS services are supported. Exadata Express is pre-configured with an Oracle Wallet that contains more than 90 of the most common root and intermediate SSL certificates. The APEX_WEB_SERVICE package automatically takes advantage of this Oracle Wallet without additional configuration from application developers. This Oracle Wallet is centrally managed and therefore you cannot consume 3rd party Web services that are protected using self-signed SSL certificates.

Exadata Express is pre-configured with a network access control list (ACL) to permit outbound Web service calls. No further configuration by application developers is necessary.

All outbound requests flow through the Oracle Cloud web proxy. The APEX_WEB_SERVICE package automatically uses this proxy without additional configuration from application developers.

There is a limit of 50,000 outbound Web service requests per Application Express workspace in a 24 hour period.

To learn more, see APEX_WEB_SERVICE in Oracle Application Express API Reference.

---

**Implement RESTful Web Services with Oracle SQL Developer**

Oracle SQL Developer is a free graphical tool that enhances productivity and simplifies database development tasks. Oracle REST Data Services (ORDS) is bundled with SQL Developer. You can use SQL Developer to implement RESTful web services in your Oracle Database Exadata Express Cloud Service.

- **Hands on Lab: AutoREST Enable Database Objects**
- **Hands on Lab: REST Development using Oracle SQL Developer**

**Topics**

- Install and Administering REST Data Services
- Auto REST Enabling Database Objects with Oracle SQL Developer
- Develop RESTful Web Services with Oracle SQL Developer

To learn more about implementing RESTful Data Services using Oracle SQL Developer, see SQL Developer User’s Guide.
Install and Administering REST Data Services

Oracle REST Data Services (ORDS) is installed and configured for all the users in the Oracle Database Exadata Express Cloud Service by default.

As ORDS is installed and configured in the Exadata Express service, all the users for this service can directly start with developing RESTful web services without performing any kind of setup. The ORDS Administration features are restricted or unavailable for Exadata Express Cloud Service users, by default.

Auto REST Enabling Database Objects with Oracle SQL Developer

AUTO Rest enable allows you to make CRUD operations via REST on your tables and views. You can use a REST API to access, change, or delete data from your tables and views, without writing any code. You simply need to turn this on by REST enabling the schema and then, the desired tables and views. The ORDS service will then honor REST calls to the database objects, returning expected responses in a JSON format to the application or website. Once you REST enable a table, you can even use a series of predicates to do filtering and sorting on that table or view.

Auto REST enable a schema

Follow these steps to enable Auto REST on your schema in the Exadata Express service:

1. Using Oracle SQL Developer, connect to your Oracle Database Exadata Express Cloud Service. See Connect SQL Developer.

   Note:
   Connect to a development schema on your Exadata Express service database, for Auto REST Enabling it.

2. Under Connections, right click your Exadata Express connection.

3. Click REST Services and select Enable REST Services.
   The RESTful Services Wizard will appear.

4. Make the following selections, and click Next.
   - Select the check-box against Enable Schema.
   - Provide a value for Schema alias. This prevents from exposing your schema name to the ORDS consumers.
   - Select check-box against Authorization required. This helps you in securing your services, and preventing unauthorized access to your services.

5. Click Finish.

Auto REST enable a table

Follow these steps to enable Auto REST a table on your schema in the Exadata Express:
1. Using Oracle SQL Developer, connect to your Oracle Database Exadata Express Cloud Service. See Connect SQL Developer.

2. Under Connections, expand your Exadata Express connection.

3. Expand the tables in your connection, and right-click the table for which you want to enable Auto REST.

4. Select Enable REST Service.
   The RESTful Services Wizard appears.

5. The RESTful Services Wizard will appear. Make the following selections, and click Next.
   - Select the check-box against Enable Object.
   - Provide a value for Object alias. This will be the name that is used in the Auto Rest URL to access this object. Therefore, this prevents exposing your object name to the ORDS consumers.
   - Select check-box against Authorization required. This helps you in ensuring that only authenticated users with the correct role can access this object.

6. Click Finish.

See Hands on Lab: AutoREST Enable Database Objects in the Getting Started with Oracle REST Data Services Series on Oracle Learning Library.

Develop RESTful Web Services with Oracle SQL Developer

The REST Data Services Development panel provides a graphical user interface for writing, testing, and publishing RESTful Services to Oracle REST Data Services. REST endpoints can be defined along with basic operations including create, query, update and delete. More complex SQL and PL/SQL operations can be defined and mapped. The ORDS service will then honor REST calls, returning expected responses in a JSON format to the application or website.

Perform the below tasks, to develop RESTful services with Oracle SQL Developer in your Oracle Database Exadata Express Cloud Service:

Note:
The steps outlined below are applicable to Oracle SQL Developer version 4.1.5. They might vary while working with any other versions above 4.1.5.

- Using Oracle SQL Developer, connect to your Oracle Database Exadata Express Cloud Service. See Connect SQL Developer.
- Create a REST development connection from Oracle SQL Developer.
  These are some of the important parameters to be entered as part of creating REST development connection for your database on Exadata Express service:
    - Connection Name: Enter any name for your new connection. The name that you provide that uniquely identifies your connection.
Implement RESTful Web Services with Oracle Application Express

RESTful Web services enable the declarative specification of RESTful services used to access the database. These services work with the Oracle REST Data Services to enable the consumption of these services.

To create a RESTful Web service:

1. Access the Oracle Application Express REST Data Services from the Oracle Database Exadata Express Cloud Service. See Access Oracle Application Express.

   The Oracle Application Express REST Data Services page appears.

2. To create a RESTful Web service, click Create.

3. Follow on-screen instructions.

To learn more about implementing RESTful Web services, see Managing RESTful Service Modules in *Oracle Application Express SQL Workshop Guide*.

Consume RESTful Web Services with JET

Oracle Java Extension Toolkit (JET) is a collection of open source JavaScript libraries along with a set of Oracle contributed JavaScript libraries that make it as simple and efficient as possible to build applications that consume and interact with Oracle products and services, especially Oracle Cloud services. Your Oracle Database Exadata Express Cloud Service allows you to consume RESTful web services using Oracle JET.
Oracle JET provides you framework to build client-side web and hybrid mobile applications based on JavaScript, HTML5, and CSS.

- Data for an Oracle JET application can come from any web data source that generates JSON data, such as a REST service, Server Sent Event (SSE), or WebSocket. In addition, Oracle JET also provides specific support for integrating web service data based on the Oracle REST standard. See About Oracle JET Data Binding in Oracle JavaScript Extension Toolkit (JET) Developing Applications with Oracle JET.

- The Oracle JET Common Model and Collection API provides a collection-of-records object model that includes classes for bringing external data into an Oracle JET application and mapping the data to the application’s view model. See Using the Oracle JET Common Model and Collection Framework in Oracle JavaScript Extension Toolkit (JET) Developing Applications with Oracle JET.

- The Oracle JET framework is designed to work with any web service that returns data in the form of JSON objects. Because the content of JSON objects can vary widely from service type, the application developer is responsible for examining the content of the JSON object and defining the ViewModel appropriately. See Integrating REST Services in Oracle JavaScript Extension Toolkit (JET) Developing Applications with Oracle JET.

Troubleshoot RESTful Web Services

If you receive “Your service is currently unavailable” error message or “500 Internal Server Error” HTTP status code when accessing RESTful web services deployed in Exadata Express, it could be caused by a problem with the corresponding handler.

For example, a PLSQL syntax error in the handler (such as a missing semicolon) or an SQL runtime exception (such as ORA-01403: no data found) are common reasons for this message. To confirm this, examine the Error-Reason header in the HTTP response using developer tools within your browser or in your REST client.

Note:
Use Base64 decoder to decode the value of the Error-Reason header.
Rapidly Developing Web Applications using Application Express

This section describes how to perform application developer tasks for Oracle Database Exadata Express Cloud Service.

Topics:
- Manage Oracle Application Express Users
- Manage Oracle Application Express Database Applications
- Manage Oracle Application Express Websheet Applications
- Provide Access to Applications
- Monitor Oracle Application Express Application Activity
- Manage Oracle Application Express Packaged Apps and Sample Code
- Manage Application Security
- Use Oracle Application Express Applications and Websheets
- Send Emails
- Application Archiving

Manage Oracle Application Express Users

You can manage your Oracle Application Express users from Oracle Application Express.

For information on managing Oracle Database Exadata Express Cloud Service users, see Adding Users and Assigning Roles in Getting Started with Oracle Cloud.

Topics
- About Oracle Application Express User Types
- Manage Oracle Application Express Users, Roles and Privileges

About Oracle Application Express User Types

In addition to Oracle Database Exadata Express Cloud Service user roles and privileges, there are Application Express user roles and privileges used to access, develop and administer Application Express applications.

See Adding Users and Assigning Roles in Getting Started with Oracle Cloud for information on Oracle Cloud user roles and privileges.
Video – Import Batch of User Accounts to your Oracle Cloud Service

Application Express has the following types of users:

- **End Users.** End users of an Oracle Application Express application managed by the Security page in My Services or Application Express authentication. Users that have been granted permission to access an Oracle Application Express application.

- **Developers.** Developers of Oracle Application Express applications. Developers have access to the Application Builder and the SQL Workshop.

- **Workspace Administrators.** Administrators given access to all Oracle Application Express application components. Additionally, they can manage application user accounts, groups and development services that use Oracle Application Express authorization.

Manage Oracle Application Express Users, Roles and Privileges

Service administrators add and manage users of Oracle Application Express applications from the Oracle Identity Console or Oracle Application Express Administration. During application creation or installation, the method for managing users is defined.

Tutorial – Managing Oracle Cloud Users and Roles Using My Services

- **Oracle Identity Console** (default). Use the Oracle Identity Console to add and manage application users if Oracle Identity Console is the application's authentication method selected during installation or creation. To learn more, see Adding Users and Assigning Roles in Getting Started with Oracle Cloud.

- **Oracle Application Express Administration.** Use the Manage Users and Groups page of the Oracle Application Express Administration to add and manage application end users if Oracle Application Express is the application's authentication method selected during installation or creation.

To manage Oracle Application Express users:

1. Access Oracle Application Express Application Builder from the Oracle Database Exadata Express Cloud Service console. See Access Oracle Application Express.

   The Application Builder home page displays.

2. Click the **Administration** menu at the top right.

   The Administration menu displays.

3. Select **Administration**.

   The Administration home page displays.
4. Click **Manage Users and Groups**.

The Manage Users and Groups page appears.

To learn more, see Managing Users in a Workspace in *Oracle Application Express Administration Guide*.

---

**Manage Oracle Application Express Database Applications**

Database applications are managed with the Oracle Application Express Application Builder. The Oracle Application Express Application Builder enables you to create, edit and remove applications.

**Topics:**

- Create a Database Application
- Modify a Database Application
- Remove a Database Application

**Create a Database Application**

The Create Wizard enables you to create a complete application containing multiple pages including reports, interactive reports, forms, tabular forms, and master detail forms. The Create Application Wizard is designed to easily and quickly create Oracle Application Express Database applications.
Follow these steps to create a Database application:

1. Access Oracle Application Express Application Builder from Oracle Database Exadata Express Cloud Service service console. See Access Oracle Application Express.
   The Application Builder home page displays.
2. Click Create.
   The first page of the Create Application wizard displays.
3. Select Desktop, or Mobile and click Next.
4. Follow on-screen instructions.

To learn more about creating Database applications, see “Creating a Database Application” in Oracle Application Express Application Builder User’s Guide.

To learn about using Database applications, see “About Database Applications” in Oracle Application Express End User’s Guide.

Modify a Database Application

Using Oracle Application Express, you can modify an existing application.

To modify a Database application:

1. Access Oracle Application Express Application Builder from the Oracle Database Exadata Express Cloud Service console. See Access Oracle Application Express.
   The Application Builder home page displays.
2. Click the Database application you want to modify.
   The Application Edit page appears.
3. Make modifications.
   To learn more about modifying Database applications, see Creating Database Applications in Oracle Application Express Application Builder User’s Guide.
   To learn about using Database applications, see About Database Applications in Oracle Application Express End User’s Guide.

Remove a Database Application

Use Oracle Application Express to remove a Database application.

To remove a Database application:

1. Access Oracle Application Express Application Builder from the Oracle Database Exadata Express Cloud Service console. See Access Oracle Application Express.
   The Application Builder home page displays.
2. Click the Database application you want to remove.
   The Application Edit page displays.
3. Under Tasks on the right panel, click Delete this Application.
4. Follow on-screen instructions.

To learn more about removing Database applications, see Deleting an Application in *Oracle Application Express Application Builder User's Guide*.

To learn about using Database applications, see About Database Applications in *Oracle Application Express End User's Guide*.

**Manage Oracle Application Express Websheet Applications**

Websheet applications are created from the Oracle Application Express Application Builder. Websheets are applications designed to be created and modified by business users, with no knowledge of SQL necessary. A Websheet is like a wiki, except that Websheets can be driven by data included in the application.

These applications are highly dynamic and defined by their users. Websheet applications include navigation controls, search capabilities, and the ability to add annotations such as files, notes, and tags. Websheet applications can be secured using access control lists and several built-in authentication models.

**Topics:**
- Create Websheet Database Objects
- Create a Websheet Application
- Modify a Websheet Application
- Remove a Websheet Application

**Create Websheet Database Objects**

Before creating a Websheet application, required database objects must be created. This section describes how to create the required objects.

To create required Websheet database objects:


2. Click **Administration** icon at top right.
   The Administration menu displays.

3. Click **Administration**.
   The Administration home page appears.

4. Under **Tasks** on the right panel, click **Websheet Database Objects**.
   The Websheet Database Objects page appears.

5. Click **Create Websheet Database Objects**.
   The required schemas and objects display.

6. Follow on-screen instructions.

To learn more, see Removing and Validating Websheet Database Objects in *Oracle Application Express Administration Guide*.
Create a Websheet Application

Before creating a Websheet, required Websheet database objects must exist.

To create a Websheet application:

2. Click Create. This first page of the Create Application wizard displays.
4. Click Create Websheet. The Websheet Created page appears.
5. To test the Websheet, click Run Websheet. To learn more about creating Websheet applications, see Create Websheet Database Objects.

To learn more about creating and modifying Websheet applications, see Creating a Websheet Application in Oracle Application Express Application Builder User's Guide. To learn about using Websheet applications, see Using Websheets in Oracle Application Express End User Guide.

Modify a Websheet Application

You can use Oracle Application Express to modify an existing Websheet application.

To modify a Websheet application:

2. Click the websheet you want to modify. The Websheet Properties page appears.
3. Make modifications.

To learn more about creating and modifying Websheet applications, see Creating a Websheet Application in Oracle Application Express Application Builder User's Guide. To learn about using Websheet applications, see Using Websheets in Oracle Application Express End User Guide.

Remove a Websheet Application

Use Oracle Application Express to remove Websheet applications from your Oracle Database Exadata Express Cloud Service.

To remove a Websheet application:

2. Click the websheet you want to remove. The Websheet Properties page appears.

3. Click **Delete**. The Confirm Delete page appears.

4. Click **Permanently Delete Now**.

To learn more about deleting Websheet applications, see Deleting a Websheet Application in *Oracle Application Express Application Builder User's Guide*.

To learn about using Websheet applications, see Using Websheets in *Oracle Application Express End User Guide*.

### Provide Access to Applications

Once an application has been created or installed to an Oracle Database Exadata Express Cloud Service instance, end users with the appropriate authorization and authentication credentials can access it using a Web browser. The application administrator simply provides the end users with the application's URL.

The following URL shows the URL an application end user would use to access an Application Express application residing within Exadata Express with an identity domain of trialaejm:


To learn more about using Application Express applications, see About Oracle Application Express in *Oracle Application Express End User Guide*.

### Monitor Oracle Application Express Application Activity

You can monitor developer activity and changes to Oracle Application Express applications within your service by accessing the Monitor Activity page in the Oracle Application Express Builder. The Monitor Activity page features links to over thirty reports that track page views, developer activity, active sessions, login attempts, environment, application errors, and workspace schemas.

To access the Monitor Activity page:


2. Click the **Administration** icon on top right. The Administration menu displays.

3. Select **Administration**. The Administration home page displays.

4. Click **Monitor Activity**.
The Monitor Activity page appears.

To learn more, see Monitoring Activity Within a Workspace in Oracle Application Express Administration Guide.

Manage Oracle Application Express Packaged Apps and Sample Code

You can install Oracle Application Express Packaged Apps and sample code.

- **Packaged Apps.** These are applications that can be installed, run, locked, unlocked and removed. They are fully supported by Oracle. You can add users, manage user access, and make changes that meet your needs.

- **Sample Code.** These are samples that illustrate a concept or demonstrate a technique. They can be installed, run, removed, viewed and modified.

Topics:

- Install Packaged Apps and Sample Code
- Lock and Unlock Packaged Apps
- Modify Packaged Apps and Sample Code
- Remove Packaged Apps and Sample Code
Install Packaged Apps and Sample Code

The Oracle Application Express Application Builder, which is available with Oracle Database Exadata Express Cloud Service, allows you to quickly install and run a variety of built-in business process applications called Packaged Apps.

Follow these steps to install a Packaged App or sample code:

1. Access Productivity Apps from the Exadata Express console. See Access Oracle Application Express.

   The Packaged Apps Gallery appears.

2. Locate and click the Packaged App or sample code you want to install.

   The application information page displays.

3. Click Install Application.

4. For Authentication, select one of the following and click Next:
   - **Oracle Cloud Identity Management** (default) - Users of this packaged application are managed through the Identity Console. See Adding Users and Assigning Roles in Getting Started with Oracle Cloud.
   - **Application Express Accounts** - Users of this packaged application are managed from the Application Express Administration. See Managing Users in a Workspace in Oracle Application Express Administration Guide.

5. On the confirmation page, click Install Application.

6. If you selected Application Express authentication, you need to add users from Oracle Application Express Administration before running the application. See
Managing Users in a Workspace in *Oracle Application Express Administration Guide*.

7. If you selected Oracle Cloud Identity Management, the users currently authorized with access to your service have access to this application. Use the Oracle Cloud portal to add users. See Adding Users and Assigning Roles in *Getting Started with Oracle Cloud*.

8. Click **Run Application** icon.

### Lock and Unlock Packaged Apps

Packaged Apps can be locked and unlocked. When locked, Packaged Apps cannot be modified.

To lock or unlock a Packaged App:

2. Locate and click the Packaged App you want to unlock or lock. The application information page displays.
3. Click **Manage**. The Manager page displays.
4. Click **Unlock** to allow modifications, or **Lock** to not allow modifications.

### Modify Packaged Apps and Sample Code

You can modify sample code and unlocked Packaged Apps just like any other Database or Websheet application you have created or imported. Packaged Apps must be unlocked before they can be modified.

See *Modify a Database Application* and *Modify a Websheet Application*.

**Note:**

Installed sample code can be modified. Installed packaged applications cannot be modified.

### Remove Packaged Apps and Sample Code

Packaged Apps and sample code can be removed from your Oracle Database Exadata Express Cloud Service using Oracle Application Express.

Follow these steps to remove a Packaged Apps and sample code:

1. Access Oracle Application Express Application Builder from the Exadata Express console. See *Access Oracle Application Express*. 
The Application Builder home page displays.

2. Locate and click the Packaged App or sample code you want to remove.
   The application information page displays.

3. Click Manage.
   The Manager page displays.

4. Click Remove.

5. To confirm, click Remove Application.

Manage Application Security

Below are some topics that describe how to implement security best practices for Oracle Application Express and Oracle Database Exadata Express Cloud Service. Administrators are primarily responsible for ensuring the security of the Oracle Application Express setup, while developers are responsible for building secure applications.

See the following topics in the *Oracle Application Express Application Builder’s Guide*:

- Managing Application Security
- Understanding Administrator Security Best Practices
- Understanding Developer Security Best Practices
- Establishing User Identity through Authentication
- Providing Security through Authorization

Use Oracle Application Express Applications and Websheets

Once an application has been created or installed to an Oracle Database Exadata Express Cloud Service instance, end users with the appropriate authorization and authentication credentials can access it using a Web browser. The application administrator simply provides the end users with the application's URL.

The following URL shows the URL of an application end user would use to access an Oracle Application Express application 100082 residing within Exadata Express with an identity domain of trialaejm:


To learn more about using Database and Websheet applications, see About Oracle Application Express in the *Oracle Application Express End User’s Guide*.

Send Emails

You can use the APEX_MAIL package to send an email from an Oracle Application Express application. APEX_MAIL contains three procedures for sending, delivering and adding attachments to emails.
Use `APEX_MAIL.SEND` to send an outbound email message from your application. Use `APEX_MAIL.PUSH_QUEUE` to deliver mail messages stored in `APEX_MAIL_QUEUE`. Use `APEX_MAIL.ADD_ATTACHMENT` to send an outbound email message from your application as an attachment.

**Example:**

```sql
BEGIN
    APEX_Mail.Send( 'to@email-address.com', 'from@email-address.com',
                    'This is some test plain text',
                    'This is some test <b>HTML</b>',
                    'Test Email Subject' );
    APEX_Mail.Push_Queue;
END;
/
```

**Note:**

There is a built in limit of 5000 emails in any given 24 hour period. If you try to send more, you get an error.

To learn more, see `APEX_MAIL` in *Oracle Application Express API Reference*.

---

### Application Archiving

Modifications to your Oracle Application Express applications can be archived to tables in your database schema. Applications that belong to Oracle Database Exadata Express Cloud Service with automatic archiving enabled are archived when they are created and when they are changed.

The Oracle APEX Application Archive packaged application manually archives applications to view, manage and restore application archives.

**Note:**

The archived application content is counted against your tablespace quota. By default, only 5 application versions are maintained. However, you can adjust this value from the Oracle APEX Application Archive packaged application. See View and Manage Application Archives.

---

**Topics:**

- Schedule Daily Automatic Application Archiving
- Manage and View Application Archives
- Manually Archiving Applications
**Schedule Daily Automatic Application Archiving**

You can schedule daily automatic archiving for all applications that belong to your Oracle Database Exadata Express Cloud Service. Automatic daily archiving is enabled and disabled from the Service Details page, under Administration. A Exadata Express enabled with application archiving generates an archive for applications when they are created and when they are changed.

You can also manually archive from the Oracle APEX Application Archive packaged application. To learn more, see Manually Archiving Applications.

To view and manage archives, use the Oracle APEX Application Archive packaged application. To learn more, see Install the Oracle APEX Application Archive Packaged App and View and Manage Application Archives.

To schedule automatic application archiving:

1. Access your Exadata Express console. See Access the Service Console.
   The My Services application appears.

2. Click Manage Application Express.
   The Manage Application Express dialog appears.

3. Click Application Archiving.
   The Application Archiving dialog appears.

   **Note:**
   If application archiving is disabled the Enable button is visible, otherwise the disable application archiving button is displayed.

4. Click Enable.

   **Note:**
   The archived application content is counted against your tablespace quota. By default, only 5 application versions are maintained. However, you can adjust this value from the Oracle APEX Application Archive packaged app. See View and Manage Application Archives.

**Manage and View Application Archives**

The Oracle APEX Application Archive packaged app views and manages archives. Application archives are performed when the Oracle Database Exadata Express Cloud Service is enabled for automatic application archiving or when a manual archive is performed.

**Topics:**

- Install the Oracle APEX Application Archive Packaged App
Run the Oracle APEX Application Archive App
View and Manage Application Archives
Restore Oracle APEX Applications

To learn more, see Schedule Daily Automatic Application Archiving and Manually Archiving Applications.

Install the Oracle APEX Application Archive Packaged App

Before you can use the Oracle APEX Application Archive packaged app, it must be installed.

To install the Application Archive packaged application:

1. Access Productivity Apps from the Oracle Database Exadata Express Cloud Service console. See Access Oracle Application Express.

The Packaged Apps Gallery appears.

2. Locate and click the APEX Application Archive app.

The application information page displays.
3. Click **Install Application**.

4. For Authentication, select one of the following and click **Next**:
   - **Oracle Cloud Identity Management** (default) - Users of this packaged application are managed through the Identity Console. See Adding Users and Assigning Roles in *Getting Started with Oracle Cloud*.
   - **Application Express Accounts** - Users of this packaged application are managed from the Application Express Administration. See Managing Users in a Workspace in *Oracle Application Express Administration Guide*.

5. On the confirmation page, click **Install Application**.

6. If you selected Application Express authentication, you need to add users from Oracle Application Express Administration before running the application. See Managing Users in a Workspace in *Oracle Application Express Administration Guide*.

7. If you selected Oracle Cloud Identity Management, the users currently authorized to have access to your Oracle Database Service have access to this application. Use the Oracle Cloud portal to add users. See Adding Users and Assigning Roles in *Getting Started with Oracle Cloud*.

8. To test the application, from the APEX Application Archive app information page, click **Run icon**.

**Run the Oracle APEX Application Archive App**

The Oracle APEX Application Archive packaged app views and manages application archives. Before you can run this application, it must be installed.

To run the Oracle APEX Application Archive packaged app:

1. Access Oracle Application Express Application Builder from the Oracle Database Exadata Express Cloud Service console. See *Access Oracle Application Express*
The Application Builder home page displays.

2. For APEX Application Archive, click the Run icon.

   The APEX Application Archive app home page appears.
   To learn how to use this application, see View and Manage Application Archives.

View and Manage Application Archives

The Oracle APEX Application Archive packaged app views and manages application archives. Before you can run this app, it must be installed.

To view and manage application archives:

1. Run the Oracle APEX Application Archive app as described in Run the Oracle APEX Application Archive App.

2. Click the Manage Archives button.
   The Archives page displays.

3. To change the maximum number of archives, under Version Retention, click Adjust.
   The Adjust Preferences page appears.

4. For Max Versions, enter the maximum number of archive versions to save, then click Apply Changes.

5. To remove archives, under Administration on left panel, click Purge All Archives.
   The Purge page appears.

6. For Purge Action, select the action to perform.

7. Click Purge Archive(s).
   To learn more, see Install the Oracle APEX Application Archive Packaged App.

Manually Archiving Applications

You can manually archive applications from the APEX Application Archive packaged app.

For further instructions on how to use this application to create an on-demand archive and schedule archives, click the help icon at the top of the APEX application Archive home page.

To manually archive applications:

1. Install the APEX Application Archive packaged app as described in Install the Oracle APEX Application Archive Packaged App.

2. Run APEX Application Archive as described in Run the Oracle APEX Application Archive App.
   The APEX Application Archive home page appears.

3. Click Archive Applications.
The Create Application Archive wizard appears.

4. Follow wizard on-screen instructions for each page to name the archive and select the applications to archive and click **Create Archive**.

The Archives page displays.

### Restore Oracle APEX Applications

The Oracle APEX Application Archive packaged app can be used to restore Oracle APEX applications.

To restore the Oracle APEX applications:

1. Once the APEX Application Archive app is installed, run the application. See **Run the Oracle APEX Application Archive App**.
2. Click **Restore**.

3. Select the archive from the Archived Content list and click **Restore**.

   You can restore workspace files by first downloading the script from Archived Content list and then import the script. See Importing an Application, Page or Component Export in **Oracle Application Express App Builder User’s Guide**.

4. On the following page, review the settings and click **Restore Content**.

   An archived copy of the application is added to the repository available to Application Builder.

5. Navigate to Oracle Application Express home page. See **Access Oracle Application Express**.

6. Click **Application Builder**.

7. Click **Workspace Utilities**.

8. Click **Export**.

9. Click **Export Repository** on the right hand navigation under Tasks.

10. Locate your file and click **Install** next to it.
Manage and Monitor Database Performance

Your Oracle Database Exadata Express Cloud Service allows you to manage and monitor its database performance both through the cloud service console and Oracle SQL Developer.

Topics

- Use the Cloud Portal to Monitor Database Performance
- Use SQL Developer to Monitor Database Performance
- Use SQL Tuning Advisor via SQL Developer for SQL Tuning

Use the Cloud Portal to Monitor Database Performance

You can monitor the performance of your database in Oracle Database Exadata Express Cloud Service from the My Services page.

Follow these steps to monitor the database performance from the Exadata Express cloud portal.

1. Access My Services from your Exadata Express service. See Access Service Details.

2. On this page, you see three tabs as:
   - **Dashboard**: Provides details of the service like service name, subscription details, current status, uptime. You also find links to view additional details of the service, and to open service console, from this tab.
   - **Users**: Provides options to manage users of this service.
   - **Notifications**: Allows you to examine system-wide and service-level notifications.
3. Click the **Exadata Express** link.

   You see complete details of the service grouped under five categories as Overview, Billing Metrics, Resource Quotas, Business Metrics, Documents, and Status.

4. Click the **Overview** tile.

   The Overview tile gives you information about the instances provisioned.

   Apart from this, the Overview tile provides you some additional information of the service like its plan, start date, subscription ID, domain and service SFTP host, port, and user name details.
5. Click **Billing Metrics**.

   This tile displays your billing information and your usage metrics. As the display is updated periodically the actual usage and invoice may differ. This metric also lists your Database instance quota and how much of it you have used. The quota is utilized when you provision a Cloud database.

6. Click the **Business Metrics**.

   This tile provides you the service usage metrics under two main categories called historical usage and latest usage. For the historical usage, you can choose a date range or duration for which you want to see the usage metrics. This tile also allows you to add a graph or export the usage information.

7. Click **Documents**.

   This option is not available at this time.

8. Click **Status**.
Provides information about the status of the service in three different views as: Monthly View, Quarterly View, and Year View. It shows a pictorial view of the service status with different color coding for different status as: Before Activation, Service Up, Planned Outage, and Service Incident.

Use SQL Developer to Monitor Database Performance

Oracle SQL Developer allows you to monitor your Oracle Database Exadata Express Cloud Service database performance in real time through different options like Real Time SQL Monitor, Active Session History (ASH) Report Writer, and Database Status.

Topics

- Use Real Time SQL Monitor in Oracle SQL Developer
- Use ASH Report Writer in Oracle SQL Developer
- View Database Status in Oracle SQL Developer

Use Real Time SQL Monitor in Oracle SQL Developer

To use the Real Time SQL Monitor, you must perform the following steps:

1. Install Oracle SQL Developer locally, and create a cloud connection from Oracle SQL Developer to your service. see Connect SQL Developer.

2. From Oracle SQL Developer, click View, and select DBA. The DBA Connections are displayed in a tree view.

3. Connect to your Exadata Express service, under the DBA connections.

4. Expand Tuning in the DBA connections tree, under your Exadata Express service connection. If you are accessing this feature under this connection, for the first time, you may see a warning message to make sure you have a valid license for the Tuning Pack with your database. Click Yes to proceed.

Note:

You must have the Tuning Pack licensed, in order to use this feature. This is because the normal licensing procured as part of subscribing your service, applies for your non-Oracle cloud databases only.
5. Click **Real Time SQL Monitor**.

The Real Time SQL Monitor tab displays details and performance metrics for all the SQL queries that are running in real time. You can browse through the pages, if there are multiple queries in the real time sql monitor.

6. Click any entry in the list displayed in the Real Time SQL Monitor.

You can see additional details for the selected entry underneath the list. It displays additional performance monitoring information for the selected entry under two tabs namely, Plan Statistics and Metrics. Plan Statistics provides you clear information like drilled down operation by line id, estimated rows, cost, executions, timeline, memory consumption, temporary memory consumption, I/O requests etc;

7. Click the **Metrics** tab.

This tab shows the real time resource consumption for different resources like CPUs, memory, I/O.

---

**Note:**

- This is refreshed in real-time, and all the information is accurate as of that moment. However, you can choose the refresh interval from the drop-down provided at the top right corner of the tab, named **Auto Refresh**.
- If you wish to view a download a snapshot of the information displayed in this tab, you can click the **Save** icon provided on the top left of the tab.

---

**View Database Status in Oracle SQL Developer**

The Database Status node in the DBA Connections tree provides Instance Viewer which provides an overview of your instance. It is a very powerful interface which provides a detailed snapshot of your instance, along with a Top SQL report.

To use the Database Status feature in Oracle SQL Developer, you must perform the following steps:

1. Install Oracle SQL Developer locally, and create a cloud connection from Oracle SQL Developer to your service. see **Connect SQL Developer**.

2. From Oracle SQL Developer, click **View**, and select **DBA**.

The DBA Connections are displayed in a tree view.
3. Connect to your Exadata Express service, under the DBA connections.

4. Expand **Database Status** in the DBA connections tree, under your Exadata Express service connection.
   
   You can see two options as:
   
   • Instance Viewer
   • Status.

5. Click **Instance Viewer**.
   
   The Instance Viewer provides you with an overview of your instance in terms of database sessions, waits, clients, processes etc; One major aspect of instance viewer is the Top SQL report that it provides along with Memory and Storage metrics.

---

**Note:**

• The Instance Viewer is refreshed in real time, and therefore you do not have a refresh option.
• The Top SQL report shown in this page can be sorted on the various attributes like, CPU Secs, Disk Reads and so on.
• You can also drill down into a top SQL item, to look into the complete query text, execution plan, SQL Tuning Advisor’s advice, and so on.

---

**Use ASH Report Writer in Oracle SQL Developer**

To view the ASH (Active Session History) Report Writer, you must perform the following steps:

1. Install Oracle SQL Developer locally, and create a cloud connection from Oracle SQL Developer to your service. see **Connect SQL Developer**.

2. From Oracle SQL Developer, click **View**, and select **DBA**.
   
   The DBA Connections are displayed in a tree view.
3. Connect to your Exadata Express service, under the DBA connections.

4. Expand **Performance** in the DBA connections tree, under your Exadata Express service connection.

5. Click **ASH Report Viewer**.

   The ASH Report Viewer page displays with links to various types of ASH reports such as: Top Events, Load Profile, Top SQL, Top PL/SQL, Top Java, Top Call Types, Top Sessions, Top Objects/Files/Latches and so on.
Use SQL Tuning Advisor via SQL Developer for SQL Tuning

Oracle SQL Developer provides SQL Tuning Advisor to tune the SQL commands in your Oracle Database Exadata Express Cloud Service.

To use the SQL Tuning Advisor from Oracle SQL Developer, you must perform the following steps:

1. Install Oracle SQL Developer locally, and create a cloud connection from Oracle SQL Developer to your Exadata Express service. See Connect SQL Developer.

2. Write the SQL query to be analyzed in the SQL Worksheet, and click the SQL Tuning Advisor...(Ctrl+F12) icon in the SQL Worksheet toolbar.
   - The SQL Tuning Advisor task is initiated based on the current statement in the SQL Worksheet.
   - Once the task completes, the SQL Tuning Advisor page opens up with performance tuning suggestions under different categories like Statistics, SQL
Profile, Indexes. For each category, you can see an overview as well as detail information.

- It may also provide a sample restructured SQL which can improve the query performance.

3. On each page, you can click the **Run SQL Script (Alt+F11)** button, if you wish to implement the recommended changes. Alternatively, you can click the **Open SQL Script Worksheet (Alt+F12)** button, if you wish to review the recommendations, and make changes before implementing it.

![SQL Tuning Advisor](image)

**Note:**

In case, you want to use the SQL Tuning Advisor from a connection other than the default administrator schema i.e; `PDB_ADMIN`, `advisor` and `administer sql tuning set` permissions must have been granted already.
Manage Database Schemas

In addition to the default database schema, created for your service automatically by the provisioning process, you can create additional database schemas and manage whether or not they are associated with Oracle Application Express.

Topics

- Find the Default Schema Name and Set the Password
- Create Database Schemas
- Manage Database Schema Association with Oracle Application Express
- Export Database Schemas to Oracle Cloud Infrastructure Object Storage Classic
- Import Database Schemas from Oracle Cloud Infrastructure Object Storage Classic
- Post Import Tasks
- Migrate Database Schema Service Schemas to Exadata Express

Find the Default Schema Name and Set the Password

The name of the default schema used by Oracle Database Exadata Express Cloud Service is called the Host Schema and can be found from the Oracle Application Express About option. After identifying the schema name, in order to start using it, you need to set the password for the schema.

To find the default schema name:

1. Go to the Oracle Application Express Application Builder home page. See Access Oracle Application Express.

   The Oracle Application Express environment displays.

2. From the Oracle Application Express home page, in the top toolbar, click the Help icon.

   The Help menu displays.

3. Click About.
The About Application Express dialog appears including the name of the default schema called Host Schema.

4. Connect with the PDB_ADMIN account, using SQL*Plus, SQLcl or SQL Developer, and issue this statement:

   ```sql
   alter user default_schema_name identified by password
   ```

   Where `default_schema_name` is the name of the default schema you found in previous steps, and `password` is the password you want to set for the schema.

Create Database Schemas

The Oracle Database Exadata Express Cloud Service has a default schema created when the service was first provisioned. You can create additional schemas for your service from the Service Console or from other database development tools, including Oracle SQL Developer, Oracle SQLcl, and Oracle SQL*Plus.

Topics
- Create a Database Schema Using the Service Console
Create a Database Schema Using the Service Console

The Service Console for Oracle Database Exadata Express Cloud Service provides an easy way to create a new schema for your service.

To create a new schema from the Service Console:

1. Access the service console for your service. See Access the Service Console.
   
   The Service Console displays.

2. Open the Manage page and click **Database Schema** under Database Management section.

   ![Database Management](image)

   The Create Database Schema dialog appears.

3. Make the following entries:
   
   a. **Schema** – enter the name of the new schema.
   
   b. **Password** – enter the password for the new schema.
   
   c. **Confirm password** – re-enter the password for confirmation.

4. Select all the following that apply:
   
   a. **Allow Access from Oracle APEX** – select to allow Oracle Application Express associate with the schema.
   
   b. **Enable Oracle REST Data Services** – select to enable Oracle REST Data Services access to the schema.

5. Click **Create Database Schema**.

   The Success dialog displays.

6. Click **Done**.
Create a Database Schema Using Oracle SQL Developer

Using Oracle SQL Developer, you can connect as a PDB_ADMIN to create a new user and create a new schema simply by executing SQL statements. A new schema is created for the new user.

To create a new schema from Oracle SQL Developer:

1. Download and install Oracle SQL Developer. See Connect SQL Developer.
4. Execute the create user statement. For example, create a new user named `smith` with a password of `password` as follows:

   ```sql
   CREATE USER smith IDENTIFIED BY password;
   ```

   **Note:**
   The password specified, after IDENTIFIED BY in this example, is needed when creating a connection to this schema from SQL Developer.

5. Grant specific access to the new schema user. For example:

   ```sql
   GRANT CREATE TABLE TO smith;
   ```

6. Verify schema creation. For example, use the following query for new user `smith`:

   ```sql
   SELECT username, account_status FROM dba_users
   WHERE username = 'SMITH';
   ```

Create a Database Schema Using Oracle SQLcl

Using Oracle SQLcl, you can connect as a PDB_ADMIN to create a new user and create a new schema simply by executing SQL statements. A new schema is created for the new user.

To create a new schema using SQLcl:

1. Download and install Oracle SQLcl. See Connect SQLcl.
2. Configure Oracle SQLcl. See Configure Oracle SQLcl Cloud Connection.
3. Connect with Oracle SQLcl. See Connect SQLcl.
4. Execute the create user statement. For example, create a new user named `smith` with a password of `password` as follows:

   ```sql
   CREATE USER smith IDENTIFIED BY password
   ```
5. Grant specific access to the new schema user. For example:

   GRANT CREATE TABLE TO smith;

6. Verify schema creation. For example, use the following query for new user smith:

   SELECT username, account_status FROM dba_users
   WHERE username = 'SMITH';

Create a Database Schema Using Oracle SQL*Plus

Using Oracle SQL*Plus, you can connect as a PDB_ADMIN to create a new user and create a new schema simply by executing SQL statements.

To create a new schema from Oracle SQL*Plus:

1. Download, install and connect with Oracle SQL*Plus. See Connect SQL*Plus.

2. Execute the CREATE USER statement. For example, create a new user named michaela with a password of password as follows:

   CREATE USER smith IDENTIFIED BY password

   Note:
   
   The password specified, after IDENTIFIED BY in this example, is needed when creating a connection to this schema from SQL Developer.

3. Grant specific access to the new schema user. For example:

   GRANT CREATE TABLE TO smith;

4. Verify schema creation. For example, use the following query for new user smith:

   SELECT username, account_status FROM dba_users
   WHERE username = 'SMITH';
Manage Database Schema Association with Oracle Application Express

For database schemas to be used by Oracle Application Express, the schema and Oracle Application Express must be associated. These types of associations can be created and removed.

Topics

• Create an Association between a Database Schema and Oracle Application Express
• Remove an Association between a Database Schema and Oracle Application Express

Create an Association between a Database Schema and Oracle Application Express

In order for a schema to be accessible from Oracle Application Express, the schema must be associated with Oracle Application Express. A schema can be associated during the schema creation process, or from the Service Console after it has been created. Default schemas are automatically associated when the service is provisioned and do not require any additional steps to use them from Oracle Application Express.

Note:

You can also associate a schema with Oracle Application Express at the time of schema creation by selecting the Allow Access from APEX option. See Create a Database Schema Using the Service Console.

To associate a schema with Oracle Application Express:

1. Go to the Manage page of the service console and click Instance Administration under Application Express section. See About Manage.

Oracle Application Express Instance Administration page appears.

2. Click Manage Workspace.

The Manage Workspace options appear.

3. Under Workspace Actions, click Manage Workspace to Schema Assignments.

The Manage Workspace to Schema Assignments page appears, listing all workspaces in your environment along with their associated schemas. This page displays as an interactive report. To customize the report, use the Search bar at the top of the page.

4. To edit an existing schema and workspace assignment, click on a workspace, select a different Workspace or Database Schema.

5. Click Apply Changes.
To manage workspace to schema assignments, see Managing Workspace to Schema Assignments in *Oracle Application Express Administration Guide*.

**Remove an Association between a Database Schema and Oracle Application Express**

Database schemas are only visible from Oracle Application Express if they are associated with Oracle Application Express. Associations can be created and removed.

To find out more about associating a Database Schema with Oracle Application Express, see Manage Database Schema Association with Oracle Application Express.

To remove an association between a schema and Oracle Application Express:

1. Go to the Manage page of the service console and click **Instance Administration** under Application Express section. See **About Manage**.
   The Oracle Application Express Instance Administration page appears.
2. Click **Manage Workspace**.
   The Manage Workspace options appear.
3. Under Workspace Actions, click **Manage Workspace to Schema Assignments**.
   To manage workspace to schema assignments, see Managing Workspace to Schema Assignments in *Oracle Application Express Administration Guide*.

---

**Export Database Schemas to Oracle Cloud Infrastructure Object Storage Classic**

You can export database cloud schemas to Oracle Cloud Infrastructure Object Storage Classic, download the `.dmp` file and import the file to an on-premises Oracle Database, or to another cloud database, or to Exadata Express.

You can export database cloud schemas to Oracle Cloud Infrastructure Object Storage Classic container from the Manage page of the Exadata Express service console. To know more about Oracle Cloud Infrastructure Object Storage Classic, see Oracle Cloud Infrastructure Object Storage Classic help center page.

To export a database schema:

1. Go to the Manage page of the Exadata Express service console and click **Export to Cloud Storage**. See **About Manage**.
   The Export to Cloud Storage window displays with a list of recent schema exports. For a new user, the list is empty.
2. Click **Create Export**.
   The Export to Cloud Storage window displays.
3. Provide the URL to the **Cloud Storage Container**. The URL is displayed on the My Services page of your Oracle Cloud Infrastructure Object Storage Classic account. For example, `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity/ContainerName`.

*foo* refers to the Oracle Cloud Identity Domain of the storage service instance. The **Create new container if it does not already exist** check-box is selected by default. Deselect the check-box if you want to export schemas to an existing folder.

The schema exports are placed in a folder called `ContainerName` that is provided at the end of the example URL.

4. Provide the **Cloud Storage User Name**. The following is the format:
5. Enter the Cloud Storage Password.

6. A default file name is displayed for Export File Name field. You can change the default name.

7. Select the schemas that you want to export from the Schemas to Export list.

8. Optionally select Include Data and click Create Export.

**Note:**

Schema exports can be overwritten if the export file name is same as an earlier existing name.

The newly exported schema appears in the list of exports on the Export to Cloud Storage window.
The process takes few minutes to export and during this period, you can cancel an export by clicking **Cancel Export** next to recent schema export appearing in the list of exports.

9. Download the export file from web console of Oracle Cloud Infrastructure Object Storage Classic by clicking on the **Actions** drop-down and selecting **Download**.

---

**See Also:**

- **Downloading an Object** in *Using Oracle Cloud Infrastructure Object Storage Classic*

---

**Import Database Schemas from Oracle Cloud Infrastructure Object Storage Classic**

You can import database cloud schemas from Oracle Cloud Infrastructure Object Storage Classic container from the Manage page of the Exadata Express service console.

**Note:**

Only schemas previously exported from Exadata Express or Database Schema Service can be imported into Exadata Express. The source and destination accounts can be from different identity domains.

To import a database schema:

1. Go to the Manage page of the Exadata Express service console and click **Import from Cloud Storage**. See **About Manage**.
   
The Import from Cloud Storage window displays with a list of recent schema imports. For a new user, the list is empty.

2. Click **Create Import**.
   
The Create Import Storage window displays.
3. Provide the URL of the Cloud Storage Container. The URL is displayed on the My Services page of your Oracle Cloud Infrastructure Object Storage Classic account. For example, https://foo.storage.oraclecloud.com/v1/Storage-myIdentity/ContainerName. foo refers to the Oracle Cloud Identity Domain of the storage service instance. The schema exports are placed in a folder called ContainerName that is provided at the end of the example URL.

Tip:

To import Database Schema Service schema, select the export file that you created using Exporting Your Database Schema in Using Oracle Database Cloud - Database Schema Service.

4. Provide the Cloud Storage User Name. The following is the format:

Storage-myIdentity:user@example.com

Tip:

For further details on the URL and format of the credentials, see Oracle Cloud Infrastructure Object Storage Classic documentation.

5. Enter the Cloud Storage Password.

6. Click Next.
This step takes the user to the Oracle Cloud Infrastructure Object Storage Classic where a list of exports are available. If you don't see the expected export files listed, verify the specified container name. To see the export files in another container, click Previous button.

7. Select the export file and click Create Import.

The newly imported schema appears in the list of imports on the Import from Cloud Storage window.
The process takes few minutes to import and during this period, you can cancel an import by clicking **Cancel Import** next to recent schema import appearing in the list of imports. Imported schemas will have a suffix added if the schema already exists.

**Post Import Tasks**

After the schemas are imported to Exadata Express, some use cases require you to perform additional steps to access data, Oracle Application Express applications, or RESTful web services.

**Topics:**
- Validate Database Privileges and Object Dependencies
- Install APEX Applications
- Restore Oracle Application Express based RESTful Web Services
- Verify Oracle REST Data Services

**Validate Database Privileges and Object Dependencies**

Describes the post import validations for regular database usage and data access.

Regardless of the source of the schema export and import, database object privileges and roles granted during normal database operation may not automatically carry over and some privileges can be missing causing invalid code. After importing, verify such grants and if necessary, manually execute again. Ensure to check for the following:

- Invalid objects
- Missing privileges

The following are specific situations where grants do not carry over. To reproduce such grants, they must be re-executed after the import operation has completed:

- If the grant was not initiated or owned by the exported schema, then the grant will not carry over when the same schema is imported.
- If another target schema that received a grant does not exist at the time of import, then the grant will not carry over successfully.
- Any grants to **PUBLIC** will not carry over.

For example, consider a scenario where **PDB_ADMIN** creates a new schema and grants **EXECUTE** privilege to this schema on the default **DBMS_CRYPTO** PL/SQL package. After exporting and importing the schema back into Exadata Express, **EXECUTE** privilege on **DBMS_CRYPTO** package can become invalid. Manually grant the privilege.
Install APEX Applications

After the schema import, you can access Oracle Application Express applications by restoring the applications using APEX Application Archive packaged app or create a new workspace and associate the schema with the newly created workspace or associate with the existing workspace.

Perform the following steps after importing database schemas to Exadata Express to access APEX applications. The following post import tasks apply only to Exadata Express customers who run APEX applications.

1. If you want to use imported database schemas with APEX applications, then associate the imported schema to a workspace. See Manage Database Schema Association with Oracle Application Express.

2. To verify and view the associated schema, login to your Oracle Application Express workspace, click SQL Workshop tab, click SQL Commands and then click the drop-down arrow next to Schema.

3. Install Oracle Application Express applications manually or execute the APEX Application Archive packaged app in Exadata Express to restore the applications archives. For more information, see Restore Oracle APEX Applications and Manage and View Application Archives.
Important:

<table>
<thead>
<tr>
<th>If...</th>
<th>And...</th>
<th>Then...</th>
</tr>
</thead>
</table>
| You are using schemas imported from Database Schema Service | -                       | • The Oracle Application Express applications are always archived on export. The applications are available to you when you import the Database Schema Service schemas.  
• Execute Step 1 through Step 3. |
| You are using schemas imported from Exadata Express | -                       | The Oracle Application Express applications are not always archived on export. Users can archive applications manually or use APEX Application Archive packaged app. See Application Archiving.  
After importing the schema, perform Step 1 and Step 3. |
| The applications were archived manually before exporting the schema | -                       | After importing the schema, perform Step 1 and migrate each application from source Exadata Express to destination Exadata Express through the import/export function in APEX. See Exporting an Application and Importing Export Files in Oracle Application Express App Builder User’s Guide. |
| The applications were not archived manually before exporting the schema | -                       |                                                                                                                                 |

Restore Oracle Application Express based RESTful Web Services

Perform the following steps after importing database schemas to restore Oracle Application Express based RESTful Web Services. These tasks apply only to Exadata Express customers who run APEX RESTful Web Services.

You can restore Oracle Application Express based RESTful Web Services by the following process:

If you want to use imported database schemas with Oracle Application Express, then associate the imported schema to a workspace. See Manage Database Schema Association with Oracle Application Express.
Important:

<table>
<thead>
<tr>
<th>If...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>You are using schemas imported from Database Schema Service</td>
<td>• The Oracle Application Express RESTful Web Services are always archived on export. The RESTful Web Services are available to you when you import the Database Schema Service schemas.</td>
</tr>
<tr>
<td></td>
<td>• Execute association of the imported schema to a workspace and restore the required RESTful Web Services.</td>
</tr>
<tr>
<td>You are using schemas imported from Exadata Express</td>
<td>1. Using APEX on the source, manually export the desired RESTful Web Services definition script. See Exporting a RESTful Service Module in Oracle Application Express SQL Workshop Guide.</td>
</tr>
<tr>
<td></td>
<td>2. Using APEX on the destination Exadata Express, import the resulting RESTful Web Service definition scripts. See Importing a RESTful Service Module in Oracle Application Express SQL Workshop Guide.</td>
</tr>
</tbody>
</table>

Verify Oracle REST Data Services

REST Data Services are defined and executed outside of the database. These services remain in force regardless of what happens to the underlying database objects they touch. However, the correct behavior can change when names of the underlying database objects they touch are changed.

This is an important consideration when importing schemas that already exist into Exadata Express. The service automatically resolves schema name conflicts by appending a number to the end of the imported schema, causing REST Data Services to continue accessing the old schema instead of the new one. In such cases, to make REST Data Services access the correct imported schema, you must take additional steps to ensure you are importing into the desired schema name. For example, you must drop an old conflicting schema first and then repeat the schema import operation.

Migrate Database Schema Service Schemas to Exadata Express

The import and export procedures for Database Schema Service is similar to that of Exadata Express procedures.

To learn more about exporting Database Schema Service schemas to Oracle Cloud Infrastructure Object Storage Classic, see Exporting Your Database Schema in Using Oracle Database Cloud - Database Schema Service. To import Database Schema
Service schemas to Exadata Express, see Import Database Schemas from Oracle Cloud Infrastructure Object Storage Classic.

\[ \text{Note:} \]

Legacy customers who purchased the Database Schema Service before August 2017, see "Creating an Oracle Data Pump Export" in Using Oracle Database Cloud - Database Schema Service. Legacy customers must manually upload the export file to an Oracle Cloud Infrastructure Object Storage Classic container before importing it into Exadata Express.
Migrate Applications and Data

Oracle Application Express applications and data can be migrated between on-premises Oracle Database installations and Oracle Database Exadata Express Cloud Service.

Topics

• About Migrating Applications
• Migrate On-premises Applications to Oracle Database Exadata Express Cloud Service
• Migrate Oracle Database Exadata Express Cloud Service Applications to On-premises Oracle Databases

About Migrating Applications

Oracle Application Express Applications and dependent database objects can be moved between Oracle Databases, whether the Oracle Database resides on-premises, or in a Database Cloud Service such as Oracle Database Exadata Express Cloud Service.

There are several different methods and tools that can be used to perform the migration. Some options include:

• Oracle Application Express Import/Export to create export files that can be imported into another Oracle Application Express workspace
• Oracle SQL Developer to drag and drop carts of data objects
• Oracle Application Express SQL Workshop to create database objects and populate them with data
• RESTful Web services to load database objects with data

See Develop on the Database.

Migrate On-premises Applications to Oracle Database Exadata Express Cloud Service

Oracle Application Express applications and data residing in an on-premises Oracle Database can be migrated to Oracle Database Exadata Express Cloud Service.

The migration process involves using Oracle Application Express, from the on-premises Oracle Database, to create an application export and generate a report that identifies application dependent database objects. The export is imported to Exadata Express, using Oracle Application Express, and the dependent objects are moved to Exadata Express using Oracle SQL Developer. See How to Move an Application to Another Instance in Oracle Application Express Application Builder User’s Guide.
From the on-premises Oracle database, create an application export from the on-premises database and find out which database objects the application needs:

1. From the on-premises Oracle Database, use Oracle Application Express to create an application export file from the on-premises Oracle Database. See Understanding the Deployment Process in Oracle Application Express Application Builder User's Guide.
   a. Go to the on-premises Oracle Application Express Builder.
      The Oracle Application Express Builder page appears.
   b. Select the application you want to migrate.
      The Application Edit page displays.
   c. Click Export / Import.
      The Import / Export dialog appears.
   d. Click Export.
      The Application Export page appears.
   e. For File Format, select the format in which you want to view the export.
   f. For Export Supporting Object Definitions, select No.
   g. Click Export.
      The application export file is created.
   h. Save the export file locally.

2. Generate a Database Object Dependencies report for the application.
   a. Go back to the Application Edit page, by clicking the application name at the start of the breadcrumb.
   b. Click Utilities.
      The Utilities page displays.
   c. Click Database Object Dependencies.
      The Database Object Dependencies page appears.
   d. Click Compute Dependencies.
      The Database Object Dependencies for this application display.
3. From Exadata Express, use Oracle Application Express to import the application export. See Importing Export Files in Oracle Application Express Application Builder User's Guide.
   a. Access Oracle Application Express in Exadata Express. See Access Oracle Application Express.  
      The Oracle Application Express home page appears.
   b. Click Application Builder.  
      The Application Builder home page displays.
   c. Click Import.  
      The Import page appears.
   d. For Choose File, navigate to the application export file you created in previous steps and select it.
   e. For File Type, select Database Application, Page or Component Export and click Next.  
      The File Import Confirmation page displays.
   f. To install now, click Next.
4. Use Oracle SQL Developer to move database objects from the on-premises Oracle Database to Exadata Express.
   a. Locate the Database Object Dependencies report you generated in an earlier step.
   b. Use Oracle SQL Developer to drag and drop required database objects, listed in the report, from the on-premises Oracle Database to Exadata Express. Follow steps outlined in About Manage.

You have completed the application migration process and are ready to run the migrated application in Exadata Express.

Migrate Oracle Database Exadata Express Cloud Service Applications to On-premises Oracle Databases

Oracle Application Express applications and data in Oracle Database Exadata Express Cloud Service can be migrated to an on-premises Oracle Database. The migration process involves using Oracle Application Express, from Exadata Express, to create an application export and generate a report that identifies application dependent database objects. The export is imported to the on-premises Oracle Database, using Oracle Application Express, and the dependent objects are moved using Oracle SQL Developer. See How to Move an Application to Another Instance in Oracle Application Express Application Builder User's Guide.
To move applications from Exadata Express to an on-premises Oracle Database:

1. From Exadata Express, use Oracle Application Express to create an application export file. See Understanding the Deployment Process in Oracle Application Express Application Builder User's Guide.
   a. Access Oracle Application Express in Exadata Express. See Access Oracle Application Express.
      The Oracle Application Express Builder page appears.
   b. Select the application you want to migrate.
      The Application Edit page displays.
   c. Click Export / Import.
      The Import / Export dialog appears.
   d. Click Export.
      The Application Export page appears.
   e. For File Format, select the format in which you want to view the export.
   f. For Export Supporting Object Definitions, select No.
   g. Click Export.
      The application export file is created.
   h. Save the export file locally.

2. Generate a Database Object Dependencies report for the application.
   a. Go back to the Application Edit page, by clicking the application name at the start of the breadcrumb.
   b. Click Utilities.
      The Utilities page displays.
   c. Click Database Object Dependencies.
      The Database Object Dependencies page appears.
   d. Click Compute Dependencies.
      The Database Object Dependencies for this application display.
This report is needed by a subsequent step to identify which data objects must be moved.

3. From the on-premises Oracle Database, use Oracle Application Express to import the application export. See Importing Export Files in Oracle Application Express Application Builder User's Guide.
   a. Go to the on-premises Oracle Application Express Builder.
      The Oracle Application Express home page appears.
   b. Click Application Builder.
      The Application Builder home page displays.
   c. Click Import.
      The Import page appears.
   d. For Choose File, navigate to the application export file you created in previous steps and select it.
   e. For File Type, select Database Application, Page or Component Export and click Next.
      The File Import Confirmation page displays.
   f. To install now, click Next.

4. Use Oracle SQL Developer to move database objects from Exadata Express to the on-premises Oracle Database.
   a. Locate the Database Object Dependencies report you generated in an earlier step.
   b. Use Oracle SQL Developer to drag and drop required database objects, listed in the report, from Exadata Express to the on-premises Oracle Database. Follow steps outlined in About Manage.

You have completed the application migration process and are ready to run the migrated application in your on-premises Oracle Database.
Manage Vanity URLs for Exadata Express Hosted Applications

Oracle Database Exadata Express Cloud Service supports vanity URLs for applications hosted on Exadata Express. You can use your own domain URL or a vanity URL to run applications on Exadata Express so that you can see a familiar name in the address bar of your browser.

Topics

• About Vanity URLs
• Configure Vanity URLs
• Remove Vanity URLs with the Exadata Express Service Console

About Vanity URLs

Exadata Express customers can promote their own brand by exposing APEX applications and REST endpoints from a custom vanity URL. This vanity URL can replace or complement the default oraclecloudapps.com domain and Oracle SSL certificate. For example, if the URL of your company is https://www.example.com, then you can expose REST APIs developed in Exadata Express using https://api.example.com vanity URL.

To configure your vanity URL, you must have a custom domain name, matching digital SSL certificate, and an instance of Oracle Cloud Infrastructure Load Balancing Classic (LBaaS). LBaaS is available through Universal Credits. You must separately obtain a load balancing instance. You can obtain a domain name from a web service provider or use the domain name of your company and obtain a digital SSL certificate from a Certificate Authority.

The vanity URL configuration is a multi-step process. The following diagram shows how LBaaS can be used to implement vanity URLs in Exadata Express.
To understand the workflow, see Workflow for Configuring Vanity URLs. In addition, you optionally can define a whitelist that restricts HTTPS access to certain IP addresses while configuring your load balancer instance.

See Also:

- Architecture Overview
- Getting Started with Oracle Cloud Infrastructure Load Balancing Classic in Using Oracle Cloud Infrastructure Load Balancing Classic
- About Universal Credits in Getting Started with Oracle Cloud
Configure Vanity URLs

The process of configuring your vanity URL can be broadly broken down to configuring an Oracle Cloud Infrastructure Load Balancing Classic (LBaaS) instance and enabling the vanity URL through the Exadata Express service console.

Topics:

- Workflow for Configuring Vanity URLs
- Obtain the Public SSL Certificate from Exadata Express
- Configure Oracle Cloud Infrastructure Load Balancing Classic Instance
- Configure Vanity URLs from the Exadata Express Service Console
- Configure an IP Whitelist Policy

Workflow for Configuring Vanity URLs

To configure your vanity URL, you must have a custom domain name, access to an Oracle Cloud Infrastructure Load Balancing Classic (LBaaS) account, and an SSL certificate. This workflow guides you through the configuration steps.

To configure your vanity URL:

1. Ensure that you have access to an Oracle Cloud Infrastructure Load Balancing Classic (LBaaS) metered account. This account is not required to be in the same Identity Domain as Exadata Express.


   Note:
   Exadata Express Cloud Service does not support self-signed certificates and unencrypted HTTP connections.

3. Obtain the public SSL certificate from your Exadata Express instance. You will use this trusted certificate to encrypt connections between your LBaaS instance and your Exadata Express instance. See Obtain the Public SSL Certificate from Exadata Express.


5. Configure the vanity URL from the Exadata Express service console. See Configure Vanity URLs from the Exadata Express Service Console.

6. Optionally, create an IP whitelist for your LBaaS instance to restrict access to your vanity URL. See Configure an IP Whitelist Policy.
Obtain the Public SSL Certificate from Exadata Express

Obtain the public SSL certificate from your Exadata Express instance. You will use this trusted certificate to encrypt connections between your LBaaS instance and your Exadata Express instance.

To obtain the public SSL certificate:

1. Access the service console for your service. See Access the Service Console.
2. Use your browser to download the public wildcard SSL certificate for *.*.db.us2.oraclecloudapps.com or *.*.db.em2.oraclecloudapps.com.
3. If your browser did not save this SSL certificate in PEM format, then you must separately convert it to PEM format. One way of converting the certificate is to use the openssl utility. For example:

   openssl x509 -inform der -in db.us2.oraclecloudapps.com.cer -out db.us2.oraclecloudapps.com.pem

   

   ✍ Note:

   Note the expiration date of this SSL certificate. Oracle renews the SSL certificate prior to the expiry date, and you must upload the new version to your LBaaS instance.

Related Topics

• Configure Oracle Cloud Infrastructure Load Balancing Classic Instance
To enable vanity URLs for your applications in Exadata Express, you must provision and configure Oracle Cloud Infrastructure Load Balancing Classic (LBaaS) instance and then enable vanity URLs through the Exadata Express service console.

Configure Oracle Cloud Infrastructure Load Balancing Classic Instance

To enable vanity URLs for your applications in Exadata Express, you must provision and configure Oracle Cloud Infrastructure Load Balancing Classic (LBaaS) instance and then enable vanity URLs through the Exadata Express service console.

To configure your LBaaS instance:

1. Provision a new LBaaS instance and make a note of the generated Canonical Host Name. See Creating a Load Balancer and About the Load Balancer IP Addresses and Canonical Host Name in Using Oracle Cloud Infrastructure Load Balancing Classic.
Note:

Oracle recommends provisioning your LBaaS instance in close geographical proximity to your Exadata Express instance. For example, for Exadata Express instances in North America data center (US2), select uscom-central-* site for the new LBaaS instance. For Exadata Express instances in EMEA data center (EM2), select eucom-north-* site for the new LBaaS instance.

2. Create the **CNAME** DNS record for your vanity domain name using the **Canonical Host Name** value. Check the documentation of your DNS registrar for specific instructions.

3. Upload the digital SSL certificate for your vanity domain name to the LBaaS instance, including all intermediate CA certificates. See Importing a Load Balancer Digital Certificate in *Using Oracle Cloud Infrastructure Load Balancing Classic*.
   - For **Certificate Type**, select **Server Certificate**.

4. Upload the trusted SSL certificate, which you obtained from Exadata Express, to the LBaaS instance. See Importing a Load Balancer Digital Certificate in *Using Oracle Cloud Infrastructure Load Balancing Classic*.
   - For **Certificate Type**, select **Trusted Certificate**.

   - For **Policy Type**, select **Trusted Certificate**.
   - For **Trusted Certificate URI**, select the public SSL certificate that you obtained from your Exadata Express instance.

   - For **Servers**, enter your Exadata Express oraclecloudapps.com domain name followed by a colon (:) and port number (443). For example, eecs-mycospany.db.us2.oraclecloudapps.com:443.

   - For **Port**, enter 443
   - For **Balancer Protocol**, select **HTTPS**
   - For **Server Protocol**, select **HTTPS**
   - For **Server Pool**, select the server pool that you created in Step 6.
   - For **Security Certificate**, select your SSL server certificate. See Obtaining a Digital Certificate in *Getting Started with Oracle Cloud*.
   - For **Policies**, select your **Trusted Certificate Policy** that you created in Step 4. See Obtain the Public SSL Certificate from Exadata Express.
   - For **Virtual Hosts**, enter your vanity domain name.

You have successfully configured an LBaaS instance. You are ready to configure your vanity URL in the Exadata Express service console.
Related Topics

- **Configure Vanity URLs from the Exadata Express Service Console**
  As a final step to enabling a vanity URL, configure it through the Exadata Express service console.

## Configure Vanity URLs from the Exadata Express Service Console

As a final step to enabling a vanity URL, configure it through the Exadata Express service console.

To configure your vanity URL:

1. Access the service console for your service. See [Access the Service Console](#).
2. Click **Manage**.

   **Note:**

   Only users with Database Administrator role in the Cloud Identity Management can see the **Manage** section.

3. Click on **Vanity URL** in the Application Express section.
   The Vanity URL dialog box appears.

4. Enter your **Vanity URL** (vanity domain name).
5. Optionally, enter a **Landing Page** URL that automatically opens when end-users access your vanity URL. For example, specify `/apex/f?p=100` to launch APEX application 100.
6. Optionally, select **Disable Application Express and REST access using the default oraclecloudapps.com URL** so that the developers and end users must utilize the vanity URL for all Application Express and REST access to this cloud database.
Tip:

Use this option in conjunction with an IP whitelist policy in your LBaaS instance. You will continue using the default oraclecloudapps.com URL for service console access regardless of this selection. To know more, see Configure an IP Whitelist Policy.

7. Click **Save Vanity URL**.

Users now have access to the vanity URL and the service console shows **Enabled**.

Related Topics

- **Configure Oracle Cloud Infrastructure Load Balancing Classic Instance**
  To enable vanity URLs for your applications in Exadata Express, you must provision and configure Oracle Cloud Infrastructure Load Balancing Classic (LBaaS) instance and then enable vanity URLs through the Exadata Express service console.

- **Remove Vanity URLs with the Exadata Express Service Console**
  You can remove vanity URLs using the Exadata Express service console.

Configure an IP Whitelist Policy

By default, your Oracle Cloud Infrastructure Load Balancing Classic (LBaaS) instance permits inbound traffic from all internet clients. Therefore, all internet clients are able to reach APEX applications and REST services hosted in your Exadata Express instance. To limit access to your Exadata Express instance only to a few trusted clients, such as your company office or your other SaaS or PaaS instances, you must define an IP whitelist policy in your LBaaS instance. This step is optional.

To configure an IP whitelist:

1. Access your LBaaS instance through the service console.
2. Click the **Policies** page.
3. Create a new **Resource Access Control Policy**.
   - For **Disposition**, select **Deny-All** to restrict access to everybody unless their IP address is explicitly defined in this policy.
   - For **Permitted Clients**, enter the list of IP addresses or CIDR ranges identifying clients to accept requests.
4. Open the **Listeners** page and edit the listener you created in Configure Oracle Cloud Infrastructure Load Balancing Classic Instance.
• For Policies, select Resource Access Control Policy created in Step 3 in addition to the existing Trusted Certificate Policy.

5. From the Exadata Express service console, open the Vanity URL dialog and select Disable Application Express and REST access using the default oraclecloudapps.com URL. Save the vanity URL configuration.

You have successfully configured an IP whitelist policy for your LBaaS instance and Exadata Express instance. Clients who are not listed in this policy receive a 403 Forbidden error when they attempt to access your vanity URL.

See Also:
Creating Policies for a Load Balancer in Using Oracle Cloud Infrastructure Load Balancing Classic

Remove Vanity URLs with the Exadata Express Service Console

You can remove vanity URLs using the Exadata Express service console.

To remove a vanity URL:
1. Access the service console for your service. See Access the Service Console.
2. Click Manage.
3. Click on Vanity URL in the Application Express section.

The Vanity URL dialog box appears.

Note:
The Vanity URL dialog now displays a Remove button.
4. Click **Remove**.

You have successfully removed the vanity URL. Changes take effect in 1 hour or less after you click the **Remove** button. All developers and end users must utilize the default oraclecloudapps.com URL for all Application Express and REST access to your cloud database.
Frequently Asked Questions (FAQ)

This section includes answers to some commonly asked Oracle Database Exadata Express Cloud Service questions.

Topics

• How do I add more database storage to Exadata Express?
• How do I track the amount of storage used?
• What is the quickest way to go directly to the service console?
• What PDB functionality is not available for this service?
• Is HTTP traffic over the internet to Exadata Express encrypted? How are SSL/TLS certificates managed?
• Can I run APEX applications on Exadata Express from our own domain name URL and SSL certificate?
• How do I create an IP whitelist for my Exadata Express instance?
• How do I stop billing in metered Exadata Express Cloud Service?
• How do I manage new Exadata Express Cloud Service instance using Universal Credits?
• Does the Exadata Express administrator need to perform Oracle Database backups?

How do I add more database storage to Exadata Express?

When you purchase a subscription for an Oracle Database Exadata Express Cloud Service, you select the shape you want. The shape you choose determines the amount of database storage provisioned for your Pluggable Database (PDB). This size is fixed for the lifetime of the service. The X20 shape has 20GB storage, while X50 and X50IM both have 50GB storage.

Before you set out to increase your database storage, it is recommended that you take full advantage of the Oracle Database compression options included with Exadata Express (such as Advanced Compression, and Hybrid Columnar Compression). These compression features often will allow you to store many more times of usable data beyond your initial 20GB or 50GB maximum.

If you determine more storage still is needed, you may decide to scale-up to a higher level of Exadata Express (for example, from X20 to X50) or to higher bands of Oracle Database Cloud Service where you get a full-instance Oracle Database running inside of a new virtual machine in Oracle Cloud. In both cases, you must copy your data from the Exadata Express PDB to the newly provisioned destination database. You can utilize the rich capabilities of Oracle SQL Developer or Oracle Application Express to copy data, or you can leverage the SQL, PL/SQL and REST interfaces to implement a custom copy operation.
How do I track the amount of storage used?

You can view the amount of storage used by your Oracle Database Exadata Express Cloud Service by going to the Service Details for your Exadata Express service.

To view the amount of used storage:

1. Go to the Service Details for your service. See Access Service Details.
   The Service Details display.

2. Under Business Metrics, you'll find the Storage Used (GB). See Understand Service Details and Metrics.
   The Storage Used value tells you the amount of storage in GB that is currently in use.

What is the quickest way to go directly to the service console?

Getting to your Oracle Database Exadata Express Cloud Service from the standard Oracle Cloud portal page involves multiple clicks and browser navigations. However, you can navigate directly to the service console with a single click, by using a bookmark.

To navigate directly to the service console with a single click:

1. Locate and click the Open Service Console link. See Access the Service Console.
2. Use it to create a new browser bookmark.
3. If it becomes necessary to update your saved bookmark, follow the service console link structure explained below:

https://[instance name]-[identity domain].db.[data center].oraclecloudapps.com/console/

- **instance name**: Name that you entered when creating the PDB using Oracle Cloud portal.
- **identity domain**: Oracle Cloud identity domain for your service account.
- **data center**: Oracle data center where your service account is provisioned. For example, use `us2` for North America.

What PDB functionality is not available for this service?

The Oracle Database Exadata Express Cloud Service is a managed Pluggable Database (PDB), in the Cloud, that shares a Container Database (CDB) with other tenants. In order to ensure the security of each tenant’s data, and overall performance of the entire Exadata Express environment, some aspects of the Oracle Database have been restricted.

For a description of the necessary boundaries that have been drawn, see the following:

- **Lockdown and Restrictions**
- **Known Issues for Oracle Database Exadata Express Cloud Service**

Is HTTP traffic over the internet to Exadata Express encrypted? How are SSL/TLS certificates managed?

Yes. HTTP traffic over the internet to Oracle Database Exadata Express Cloud Service is encrypted by SSL/TLS. Unencrypted access is not permitted.

Exadata Express uses public SSL/TLS certificates signed by DigiCert certificate authority for all HTTP access. These certificates are trusted by modern web browsers by default and are renewed by Oracle annually. For non-browser clients that connect to APEX apps and REST endpoints deployed in Exadata Express, Oracle recommends adding the DigiCert Global Root CA certificate to the client’s trust store (for example, Oracle Wallet, Java TrustStore). When this root certificate is trusted, annual certificate renewals are transparent to the client.

Alternatively, to gain full control of the SSL/TLS certificate presented by your Exadata Express instance, customers can set up a custom domain name and certificate. See [Manage Vanity URLs for Exadata Express Hosted Applications](#) for more information.
Can I run APEX applications on Exadata Express from our own domain name URL and SSL certificate?

APEX applications running on Exadata Express are located at the following domain name: oraclecloudapps.com. The applications use Oracle’s SSL certificate.

To configure your own domain URL and SSL certificate, see Manage Vanity URLs for Exadata Express Hosted Applications.

How do I create an IP whitelist for my Exadata Express instance?

You can define a whitelist that restricts HTTPS access to certain IP addresses while configuring a vanity URL for your Exadata Express instance. The optional whitelisting policy can be defined in the Oracle Cloud Infrastructure Load Balancing Classic (LBaaS) instance used in this configuration.

To know more, see Manage Vanity URLs for Exadata Express Hosted Applications and Configure an IP Whitelist Policy.

How do I stop billing in metered Exadata Express Cloud Service?

Currently, the only way to stop or pause billing in metered Oracle Database Exadata Express Cloud Service is to delete the service instance (PDB) from My Services. Locking the instance in My Services, or executing any `ALTER PLUGGABLE DATABASE` statements does not stop the meter. Customers are advised to export their data to Oracle Cloud Infrastructure Object Storage Classic with their valid Oracle Cloud Infrastructure Object Storage Classic account using Service Console prior to deleting the service instance.

Related Topics

- **Access My Services**
  The My Services page lists all of the service consoles you have access to.

- **Export Database Schemas to Oracle Cloud Infrastructure Object Storage Classic**
  You can export database cloud schemas to Oracle Cloud Infrastructure Object Storage Classic, download the `.dmp` file and import the file to an on-premises Oracle Database, or to another cloud database, or to Exadata Express.

How do I manage new Exadata Express Cloud Service instance using Universal Credits?

Customers with Oracle Identity Cloud Service (IDCS) cloud account using Universal Credits to pay do not have the ability to create and manage Oracle Database Exadata Express Cloud Service instances while logged in with IDCS.
You must switch over to Traditional account by clicking on the drop-down next to the **Cloud Account** field on the My Services page. Alternately, you can click the Traditional Cloud Account URL that you receive in the Oracle Database Exadata Express Cloud Service welcome e-mail that takes you to the login page for traditional account. For information on how to switch to Traditional account, see Signing In to Your Traditional Cloud Account in *Getting Started with Oracle Cloud*.

### Does the Exadata Express administrator need to perform Oracle Database backups?

Customers must implement their own backup strategy.

For example, Exadata Express customers can click **Export to Cloud Storage** from the *Manage* page of the service console to create their own backups. These backups can be restored later using the **Import from Cloud Storage** feature.

---

**Note:**

Oracle Cloud Operations create internal backups of Exadata Express regularly. These backups restore internal systems in the unlikely event of a severe outage impacting multiple customers.

---

**Related Topics**

- **Export Database Schemas to Oracle Cloud Infrastructure Object Storage Classic**
  You can export database cloud schemas to Oracle Cloud Infrastructure Object Storage Classic, download the `.dmp` file and import the file to an on-premises Oracle Database, or to another cloud database, or to Exadata Express.

- **Import Database Schemas from Oracle Cloud Infrastructure Object Storage Classic**
  You can import database cloud schemas from Oracle Cloud Infrastructure Object Storage Classic container from the *Manage* page of the Exadata Express service console.
Lockdown and Restrictions

To ensure the security of each tenant's data, and the overall performance integrity of the entire Oracle Database Exadata Express Cloud Service environment, some aspects of the Oracle Database have to be restricted.

The limitations required to protect security and performance integrity are detailed in this section. None of the limitations listed have been put in place as an attempt to limit the functionality of the Exadata Express. Virtually all standard SQL and PL/SQL syntax and constructs used with the Oracle Database work in this service.

For additional restrictions, see Feature Restrictions and Limitations in Known Issues for Oracle Database Exadata Express Cloud Service.

Topics:

- Lockdown Profile
- Resource Restrictions
- Simultaneous Connection Considerations

Lockdown Profile

A PDB lockdown profile is used to restrict the features and options available to Oracle Database Exadata Express Cloud Service users. These restrictions are important to enhance security.

There is a default lockdown profile for each type of Exadata Express depending on the size of the PDB (X20, X50, and X50IM). The service uses the Resource Manager to limit resources used by each tenant in the service. This is complementary to the lockdown profile that is used for each tenant. See Resource Restrictions for further information.

The following lists the security restrictions imposed on Exadata Express by the lockdown profile:

- **ALTER SYSTEM** is disabled except **ALTER SYSTEM SET**. In other words, you cannot issue a **ALTER SYSTEM** statement unless it contains the **SET** clause as well.

- **ALTER SYSTEM SET** is allowed except for the following cases:
  - It cannot be used to alter the value of any hidden parameters (underscore parameters).
  - It cannot be used to set **SQL_TRACE**, **EVENTS**, **ENABLE_DDL_LOGGING**, **07_DICTIONARY_ACCESSIBILITY**, and **SEC_PROTOCOL_ERROR_TRACE_ACTION**.

- **ALTER SESSION** is disabled if you try to:
  - Alter the value of any hidden parameters (underscore parameters).
  - Set **SQL_TRACE**, **EVENTS**, **ENABLE_DDL_LOGGING**.
• All ALTER PLUGGABLE DATABASE and ALTER DATABASE statements are disabled except the ones that do the following operations:
  – Set DEFAULT EDITION, DEFAULT TEMPORARY TABLESPACE, TABLESPACE, TIME_ZONE, {DATAFILE|TEMPFILE} RESIZE, {DATAFILE|TEMPFILE} AUTOEXTEND ON, {DATAFILE|TEMPFILE} AUTOEXTEND OFF
  – {OPEN|CLOSE}
• Operating system access, common schema access, and AWR access are all disabled.
• Network access is disabled, except when using APEX_WEB_SERVICE and APEX_MAIL PL/SQL APIs. See:
  – "APEX_WEB_SERVICE" in Application Express API Reference.
  – "APEX_MAIL" in Application Express API Reference.

Resource Restrictions

Resources used by each Oracle Database Exadata Express Cloud Service are controlled and restricted by the Oracle Resource Manager. This section describes the resource restrictions imposed on each Exadata Express Pluggable Database (PDB), depending on the service shape and pricing model.

Table A-1 Settings for Metered Pricing Model

<table>
<thead>
<tr>
<th>Resource</th>
<th>X20</th>
<th>X250</th>
<th>X500</th>
<th>X1000</th>
<th>X1000IM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max CPU</td>
<td>1 core</td>
<td>1 core</td>
<td>2 core</td>
<td>4 core</td>
<td>4 core</td>
</tr>
<tr>
<td>CPU_COUNT (Number of CPU threads this PDB is instance caged to)</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>SESSIONS</td>
<td>60</td>
<td>60</td>
<td>120</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>Storage</td>
<td>20G</td>
<td>250G</td>
<td>500G</td>
<td>1000G</td>
<td>1000G</td>
</tr>
<tr>
<td>PGA_AGGREGATE_LIMIT</td>
<td>3.5G</td>
<td>3.5G</td>
<td>7.5G</td>
<td>15G</td>
<td>15G</td>
</tr>
<tr>
<td>PGA_AGGREGATE_TARGET</td>
<td>1.75G</td>
<td>1.75G</td>
<td>3.75G</td>
<td>7.5G</td>
<td>7.5G</td>
</tr>
<tr>
<td>SGA_TARGET</td>
<td>4G</td>
<td>4G</td>
<td>7.5G</td>
<td>15G</td>
<td>25G</td>
</tr>
<tr>
<td>Data Transfer (Per month)</td>
<td>120G</td>
<td>1500G</td>
<td>3000G</td>
<td>6000G</td>
<td>6000G</td>
</tr>
<tr>
<td>INMEMORY_SIZE</td>
<td>not available</td>
<td>not available</td>
<td>not available</td>
<td>not available</td>
<td>10G</td>
</tr>
</tbody>
</table>

Table A-2 Settings for Non-Metered Pricing Model

<table>
<thead>
<tr>
<th>Resource</th>
<th>X20</th>
<th>X50</th>
<th>X50IM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max CPU</td>
<td>1 core</td>
<td>1 core</td>
<td>1 core</td>
</tr>
</tbody>
</table>
Table A-2  (Cont.) Settings for Non-Metered Pricing Model

<table>
<thead>
<tr>
<th>Resource</th>
<th>X20</th>
<th>X50</th>
<th>X50IM</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU_COUNT</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(Number of CPU threads this PDB is Instance Caged to)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SESSIONS</td>
<td>60</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>Storage</td>
<td>20G</td>
<td>50G</td>
<td>50G</td>
</tr>
<tr>
<td>PGA_AGGREGATE_LIMIT</td>
<td>3G</td>
<td>5G</td>
<td>5G</td>
</tr>
<tr>
<td>PGA_AGGREGATE_TARGET</td>
<td>1.5G</td>
<td>2.5G</td>
<td>2.5G</td>
</tr>
<tr>
<td>SGA_TARGET</td>
<td>3G</td>
<td>5G</td>
<td>10G</td>
</tr>
<tr>
<td>Data Transfer (Per month)</td>
<td>120G</td>
<td>300G</td>
<td>300G</td>
</tr>
<tr>
<td>INMEMORY_SIZE</td>
<td>not available</td>
<td>not available</td>
<td>5G</td>
</tr>
</tbody>
</table>

Note:
Non-metered Exadata Express shapes are available only to the existing non-metered customers. All new Exadata Express customers must choose from the shapes available in the metered pricing model.

Related Topics
- Manage and Monitor Database Performance
  Your Oracle Database Exadata Express Cloud Service allows you to manage and monitor its database performance both through the cloud service console and Oracle SQL Developer.

Simultaneous Connection Considerations

When connections are being established simultaneously over Oracle Net Services (SQL*Net) to Oracle Database Exadata Express Cloud Service, there are certain parameter settings to consider in order to avoid intermittent errors during connection establishment.

Intermittent Error
If you are trying to establish a number of simultaneous connections, which may be from the use of a connection pool, or there are transient load scenarios, you may intermittently encounter the following error: ORA-28865: SSL connection closed.

Workaround
You may reduce the initial pool size and/or minimum pool size to reduce the number of simultaneous connection requests. The specific parameters depend on the pool being used by the application.
You can also add RETRY_COUNT and RETRY_DELAY parameters to the connect string in tnsnames.ora if 12.1.0.2 (or higher) version of client is used. Please start with a value of 1, and tune upwards as needed. See RETRY_COUNT and RETRY_DELAY in Oracle Database Net Services Reference.

If the issue persists, contact Oracle Support for assistance.