

Oracle® REST Data Services

Installation and Configuration Guide



Release 23.4

F89891-04

May 2024

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Copyright © 2011, 2024, Oracle and/or its affiliates.

Primary Authors: Mamata Basapur, Chuck Murray, Tulika Das

Contributors: Kris Rice, Jeff D. Smith, Colm Divilly, Peter J. Obrien, Dermot O'Neill, Elizabeth Saunders, Ashley Chen, Sharon Kennedy, Ganesh Pitchaiah, Jason Straub, Vladislav Uvarov

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	xi
Documentation Accessibility	xi
Related Documents	xi
Conventions	xi

1 Oracle REST Data Services Installation Checklist

1.1	System Requirements	1-1
1.1.1	GraalVM Configuration	1-2
1.2	Supported Java EE Application Servers	1-2
1.3	Supported Oracle APEX Versions	1-2
1.4	ORDS Installer Privileges Script	1-3
1.5	Privileges Granted to the Oracle REST Data Services Users	1-3
1.6	About the Database Users Used by Oracle REST Data Services	1-5
1.7	About Command-Line Interface	1-6
1.8	About Online Help	1-6

2 Installing and Configuring Oracle REST Data Services

2.1	Installing Oracle REST Data Services	2-1
2.1.1	Downloading ORDS	2-2
2.1.2	Setting Up the Configuration Folder Structure	2-3
2.1.2.1	Specifying the New ORDS Configuration Folder	2-3
2.1.3	Interactive Command-Line Interface Installation	2-5
2.1.3.1	Entering a Number to Select the Type of Installation	2-9
2.1.3.2	Entering a Number to Select the Database Pool	2-9
2.1.3.3	Entering the Database Pool Name	2-10
2.1.3.4	Entering a Number to Select the Database Connection	2-10
2.1.3.5	Entering the Database Host Name	2-10
2.1.3.6	Entering the Database Listener Port	2-10
2.1.3.7	Entering the Database Service Name	2-11
2.1.3.8	Entering the TNS Location	2-11
2.1.3.9	Entering a Number to Select the TNS Network Alias	2-11

2.1.3.10	Entering the Custom Database URL	2-11
2.1.3.11	Entering the Administrator Username	2-11
2.1.3.12	Entering the Database Password	2-11
2.1.3.13	Entering the Default Tablespace	2-11
2.1.3.14	Entering a Number to Select Additional Feature(s)	2-11
2.1.3.15	Entering a Number to Configure and Start ORDS in Standalone Mode	2-12
2.1.3.16	Entering a Number to Use HTTP or HTTPS Protocol	2-12
2.1.3.17	Entering the HTTP Port	2-12
2.1.3.18	Entering the HTTPS Port	2-12
2.1.3.19	Entering a Number to Select the Certificate Type	2-12
2.1.3.20	Entering the SSL Hostname	2-12
2.1.3.21	Entering the Path for the SSL Certificate	2-12
2.1.3.22	Entering the Path for the SSL Certificate's Private Key	2-12
2.1.3.23	Entering the APEX Static Resources Location	2-13
2.1.4	Non-Interactive Command-Line Interface Installation (Silent)	2-13
2.1.4.1	Understanding Command Options for Command-Line Interface Installation	2-13
2.1.4.2	Using Input Redirection	2-16
2.2	Repairing the Oracle REST Data Services Installation	2-17
2.2.1	Command Options for Repair CLI	2-18
2.3	Upgrading Oracle REST Data Services	2-19
2.3.1	Migrate Configuration and Upgrade from ORDS 21.4.x or Earlier Releases	2-20
2.3.1.1	Interactive Upgrade	2-20
2.3.1.2	Silent Upgrade	2-23
2.3.1.3	Upgrading the Multiple Database Pools	2-24
2.4	Uninstalling Oracle REST Data Services	2-24
2.4.1	Interactive Uninstall CLI	2-24
2.4.1.1	Entering a Number to Select the Database Pool	2-26
2.4.1.2	Entering a Number to Select the Database Connection Type	2-26
2.4.1.3	Entering the Database Host Name	2-26
2.4.1.4	Entering the Database Listener Port	2-26
2.4.1.5	Entering the Database Service Name	2-27
2.4.1.6	Entering the TNS Location	2-27
2.4.1.7	Entering a Number to Select the TNS Network Alias	2-27
2.4.1.8	Entering the Custom Database URL	2-27
2.4.1.9	Entering the Administrator Username	2-27
2.4.1.10	Entering the Database Password for SYS AS SYSDBA	2-27
2.4.1.11	Entering an Option for Uninstalling ORDS	2-27
2.4.2	Non-Interactive Uninstall CLI	2-27
2.4.2.1	Using Input Redirection	2-28
2.4.2.2	Command options for Uninstall CLI	2-29
2.5	Updating the Configuration Settings	2-29

3	Configuring Oracle REST Data Services for Multiple Databases	
3.1	About the Request URL	3-1
3.2	Configuring Additional Databases	3-1
3.3	Advanced Routing	3-2
3.4	Routing Based on the Request Path	3-3
3.5	Routing Based on the Request Hostname	3-3
4	Overview of Schema Level Configuration Override	
4.1	Features of Schema Level Configuration Override	4-1
4.2	Overridable Configuration Properties	4-1
4.3	Use Cases	4-2
4.4	Scope and Visibility Matrix	4-3
5	Miscellaneous Configuration Options of Oracle REST Data Services	
5.1	Using the REST Enabled SQL Service with MySQL Database	5-1
5.1.1	Understanding the Database Credentials Source Setting	5-1
5.1.2	Configuring a Pool for MySQL Database	5-2
5.1.2.1	Configuring ORDS for Supported Containers	5-3
5.2	Configuring Jetty in ORDS Standalone Mode	5-5
5.2.1	Using Java System Properties	5-6
5.2.2	Java System Properties	5-6
5.2.3	Using Jetty XML Configuration Files	5-7
5.3	Support for Oracle RAC Fast Connection Failover	5-9
5.4	Configuring ORDS with Kerberos Setup	5-10
5.5	Authorizing Oracle REST Data Services to Access Oracle Data Guard Protected Users	5-11
5.6	Configuring REST-Enabled SQL Service Settings	5-11
5.7	Configuring the Maximum Number of Rows Returned from a Query	5-11
5.8	Configuring ICAP Server Integration for Virus Scan	5-12
5.9	Configuring the Custom Error Pages	5-12
5.10	Managing ORDS Administrator Privilege	5-13
5.10.1	Provisioning ORDS_ADMINISTRATOR_ROLE to a User	5-13
5.10.2	Unprovisioning ORDS_ADMINISTRATOR_ROLE from a User	5-13
5.11	Managing ORDS Runtime Privilege	5-14
5.11.1	Provisioning ORDS_RUNTIME_ROLE to a User	5-14
5.11.2	Unprovisioning ORDS_RUNTIME_ROLE from a User	5-15
5.12	Using OAuth2 in Non-HTTPS Environments	5-15
5.13	Configuring ORDS Metadata Cache	5-16

6 Deploying and Monitoring Oracle REST Data Services

6.1	Deploying Oracle REST Data Services	6-1
6.1.1	Serve Commands for Running in Standalone Mode	6-1
6.1.1.1	Non-Interactive Serve CLI	6-1
6.1.1.2	Serve Command Options	6-2
6.1.1.3	Converting a Private Key to DER (Linux and Unix)	6-3
6.1.1.4	Configuring a Doc Root for Non-APEX Static Resources	6-3
6.1.2	Oracle WebLogic Server	6-4
6.1.2.1	About Oracle WebLogic Server	6-5
6.1.2.2	Configuring Oracle APEX Images	6-5
6.1.2.3	Launching the Administration Server Console	6-6
6.1.2.4	Deploying ORDS on Oracle WebLogic Server	6-7
6.1.2.5	Configuring WebLogic to Handle HTTP Basic Challenges Correctly	6-9
6.1.2.6	Verifying the State and Health of ords and i	6-10
6.1.3	Apache Tomcat	6-10
6.1.3.1	About Apache Tomcat	6-10
6.1.3.2	Configuring Oracle APEX Images	6-10
6.1.3.3	Deploying ORDS on Apache Tomcat	6-11
6.1.4	Oracle Cloud Infrastructure	6-12
6.1.4.1	About Oracle Cloud Infrastructure	6-12
6.1.4.2	Configuring ORDS for High Availability on Oracle Cloud Infrastructure	6-12
6.1.4.3	Advantages of Deploying ORDS with High Availability on Oracle Cloud Infrastructure	6-13
6.2	Monitoring Oracle REST Data Services	6-13
6.2.1	Using the Instance API	6-13
6.2.1.1	Enabling the ORDS Instance API	6-14
6.2.1.2	Authorization for Using the ORDS Instance API	6-14
6.2.1.3	API Document	6-14
6.2.2	Using OpenTelemetry with Oracle REST Data Services	6-14
6.2.3	ORDS CLI Commands for Managing Pools Using JMX	6-17

7 Using the Multitenant Architecture with Oracle REST Data Services

7.1	Setting Up ORDS in a CDB Environment	7-1
7.1.1	Installation Enabling Multiple Releases	7-2
7.1.1.1	Command Line Installation	7-2
7.1.1.2	Advanced Installation	7-2
7.1.1.3	Silent Installation	7-3
7.1.2	Upgrading Oracle REST Data Services in a CDB Environment	7-3
7.1.2.1	Migrating Oracle REST Data Services in the CDB to Enable Multiple Releases	7-4
7.1.3	Uninstalling Oracle REST Data Services in a CDB Environment	7-4

7.2	Setting Up ORDS in an Application Container	7-4
7.2.1	Prerequisites for Creating ORDS in an Application Container	7-5
7.2.1.1	Creating an Application Root Container	7-5
7.2.2	Installing ORDS in the Application Root Container	7-6
7.2.3	Creating an Application Seed	7-7
7.2.4	Creating an Application PDB from the Application Seed	7-8
7.2.5	Configuring the PLSQL Gateway for APEX	7-8
7.2.6	ORDS Configuration Files Setup	7-9
7.2.6.1	Specifying the ORDS Configuration Folder	7-9
7.2.6.2	Creating the ORDS Configuration Files for the Application Root Container	7-9
7.2.7	Running ORDS	7-11
7.2.8	Upgrading ORDS in the Application Container	7-11
7.2.9	Uninstalling ORDS from the Application Container	7-12
7.2.10	Verifying ORDS in the Application Container	7-13
7.3	Making All PDBs Addressable by Oracle REST Data Services (Pluggable Mapping)	7-13

8 Installing and Configuring Customer Managed ORDS on Autonomous Database

8.1	About Customer Managed Oracle REST Data Services on Autonomous Database	8-1
8.2	Preinstallation Tasks	8-2
8.2.1	Downloading Wallet	8-2
8.2.2	Oracle APEX Static Resources	8-2
8.3	ORDS Command-Line Interface for Customer Managed ORDS	8-3
8.3.1	Interactive Install for ADB Command Line Interface	8-3
8.3.1.1	Customer Managed ORDS Command Options	8-4
8.3.1.2	Interactive Installation Prompts	8-5
8.3.2	Silent Installation of ADB on Command-Line Interface	8-10
8.3.2.1	Using Input Redirection	8-11

9 Oracle API for MongoDB Support

9.1	Getting Started	9-2
9.2	Requirements	9-4
9.3	Configurable Settings for MongoDB	9-4
9.4	Examples	9-5
9.5	Accessing the Connection Pools	9-6
9.6	Logging MongoDB API Access	9-7
9.7	Achieving High Performance	9-7

10 Migration of Legacy Configuration

10.1	Migrating From Legacy Configurations Using the CLI Installation	10-1
10.2	Migrating URL Mapping	10-2

A Non-Interactive ORDS CLI Commands Examples

A.1	Non-Interactive Install/Upgrade	A-1
A.2	Non-Interactive Install/Upgrade (create a proxied PLSQL Gateway User)	A-1
A.3	Non-Interactive Install/Upgrade (specify PLSQL Gateway User and not using RESTful Services)	A-2
A.4	Non-Interactive Install/Upgrade (disable a configured PLSQL Gateway)	A-2
A.5	Non-Interactive Uninstall ORDS	A-2
A.6	config CLI Examples	A-3

B ORDS Legacy Commands and New ORDS CLI Commands

C About the Oracle REST Data Services Configuration Files

C.1	Understanding the Configuration Folder Structure	C-1
C.2	Understanding the Configuration File Format	C-2
C.3	Understanding the Configurable Settings	C-2

Index

List of Examples

5-1	Using the JDK_JAVA_OPTIONS	5-6
5-2	Using a specific access log format	5-8
5-3	Always returning a certain header in the response	5-8
5-4	Configuring custom error page for “HTTP 404” status code	5-13
5-5	Using Grant command	5-13
5-6	Using ORDS_ADMIN package method	5-13
5-7	Using REVOKE command	5-14
5-8	Using ORDS_ADMIN package method	5-14
5-9	Using Grant command	5-14
5-10	Using ORDS_ADMIN package method	5-14
5-11	Using REVOKE command	5-15
5-12	Using ORDS_ADMIN package method	5-15
7-1	Creating ORDS Configuration	7-9

List of Tables

2-1	Interactive Installation Prompts	2-6
2-2	Command Options for Command-Line Interface Installation	2-13
2-3	Command Options for Repair CLI	2-18
2-4	Interactive Upgrade Commands	2-21
2-5	Interactive Uninstall Prompts	2-25
2-6	Command Options for Uninstall CLI	2-29
4-1	Overidable Configuration Properties	4-2
4-2	Use Cases of Schema Level Configuration Override feature	4-3
4-3	Scope and Visibility Matrix	4-4
5-1	Java System Properties	5-6
5-2	Configuration Properties for ORDS Metadata Cache	5-16
8-1	Interactive Installation Prompts	8-6
9-1	Requirements	9-4
9-2	Configuration Settings to Support the Mongo API	9-5
B-1	ORDS Legacy and New CLI Commands	B-1
C-1	Oracle REST Data Services Configuration Settings	C-3

Preface

Oracle REST Data Services Installation, Configuration, and Development Guide explains how to install and configure Oracle REST Data Services. (Oracle REST Data Services was called *Oracle Application Express Listener* before Release 2.0.6.)

Topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This document is intended for system administrators or application developers who are installing and configuring Oracle REST Data Services. This guide assumes you are familiar with web technologies, especially REST (Representational State Transfer), and have a general understanding of Windows and UNIX platforms.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information and resources relating to Oracle REST Data Services, see the following the Oracle Technology Network (OTN) site:

<http://www.oracle.com/technetwork/developer-tools/rest-data-services/>

Conventions

The following text conventions are used in this document:

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

1

Oracle REST Data Services Installation Checklist

This chapter provides information about supported platforms, system requirements, and Database privileges required for using ORDS .

1.1 System Requirements

Oracle REST Data Services system requirements are as follows:

- Oracle Database (Enterprise Edition, Standard Edition or Standard Edition One) release 11g Release 2 or later, or Oracle 21c Express Edition.
- Use one of the following:
 - Oracle Java version 11 or 17
 - Oracle GraalVM Enterprise Edition for Java version 11 or 17

Note:

On Linux system, if the Oracle Java version is lower than 11, then the installation fails with an error message. For example:

```
Error: ORDS requires Java 11 and above to run.  
Found Java version 1.  
Please set JAVA_HOME to appropriate version and update PATH  
if necessary.
```

So, you must set `JAVA_HOME` to the appropriate version and update the `PATH` if required.

On Windows system, the user is re-directed to the Java download page.

- Web browser requirements: Refer to [Oracle Software Web Browser Support Policy](#) for more information.

Note:

Oracle APEX is *not* a prerequisite for using Oracle REST Data Services.

If Oracle APEX is installed and if RESTful services have been configured during the installation (see the step [Configuring Oracle REST Data Services in Oracle Application Express Installation Guide](#)), then Oracle REST Data Services supports it.

1.1.1 GraalVM Configuration

To operate, Oracle REST Data Services requires a Java Runtime Environment. Customers can choose to use a supported Oracle GraalVM for improved performance in their runtime environment. It is not mandatory.

However, certain features in ORDS require the Java Runtime Environment to be a GraalVM with specific Graal components installed. The dependent Graal component for such features is mentioned in the documentation section. The most common dependency is on the `Graal.js` component for JavaScript support.

Use `gu list` command to determine the components available:

```
gu list
ComponentId          Version          Component name
Stability
-----
-----
graalvm              23.0.1          GraalVM Core
Supported
icu4j                23.0.1          ICU4J
Supported
js                   23.0.1          Graal.js
Supported
regex                23.0.1          TRegex
Supported
```

If the JavaScript component is not listed as an available component, then install it using the `gu install js` command. Refer to GraalVM documentation for more information on the Graal Updater command utility and the capabilities of the various Graal components.

1.2 Supported Java EE Application Servers

Oracle REST Data Services supports the following Java EE application servers:

Application Server	Supported Release
Oracle WebLogic Server	14c Release and later
Apache Tomcat	Release 8.5.x through Release 9.0.x



Note:

ORDS cannot run on Tomcat Release 10.

1.3 Supported Oracle APEX Versions

Oracle REST Data Services supports the currently supported versions of APEX.

 **See Also:**

The Oracle APEX (Formerly HTML DB) table in the ORACLE INFORMATION-DRIVEN SUPPORT document for supported versions of APEX.

1.4 ORDS Installer Privileges Script

This section describes about the script file that provides privileges to the user to install, upgrade, repair, and uninstall ORDS.

 **Note:**

This script is used when you do not want to use SYS AS SYSDBA to install, upgrade, repair, and uninstall ORDS for Oracle PDB or Oracle 11g.

The Oracle REST Data Services contains a script, `ords_installer_privileges.sql` which is located in the `scripts/installer` folder. The script provides the assigned database user the privileges to install, upgrade, repair, and uninstall ORDS in Oracle Database.

Perform the following steps:

1. Using SQLcl or SQL*Plus, connect to the Oracle Database instance. You must have a Database account with appropriate privileges for installing ORDS.
2. Execute the following script providing the database user:

```
SQL> @/path/to/scripts/installer/ords_installer_privileges.sql exampleuser
SQL> exit
```

You must use the specified database user to install, upgrade, repair, and uninstall ORDS.

1.5 Privileges Granted to the Oracle REST Data Services Users

As part of the Oracle REST Data Services installation, privileges are granted to several users and roles:

- **ORDS_RUNTIME_ROLE role**
 - **ORDS_RUNTIME_ROLE is granted EXECUTE on the following packages if these packages are not granted EXECUTE to PUBLIC:**
 - * SYS.DBMS_LOB
 - * SYS.DBMS_SESSION
 - * SYS.DBMS_UTILITY
 - * SYS.WPIUTL
 - **ORDS_RUNTIME_ROLE is granted the necessary ORDS_METADATA object privileges to determine the repository version and to access the connection pool configurations.**
- **ORDS_PUBLIC_USER user**
 - **ORDS_PUBLIC_USER is granted connect to allow connection to the database.**

- ORDS_PUBLIC_USER is granted role, ORDS_RUNTIME_ROLE to allow the user to act as an ORDS runtime user
- ORDS_ADMINISTRATOR_ROLE role
 - ORDS_ADMINISTRATOR_ROLE is granted EXECUTE on ORDS_METADATA.ORDS_ADMIN PL/SQL package.
- PUBLIC
 - PUBLIC is granted EXECUTE on ORDS_METADATA.ORDS_REPVERSION view to allow the repository version to be queried by anyone.
 - PUBLIC is granted SELECT on many ORDS_METADATA views.
 - PUBLIC is granted EXECUTE on ORDS_METADATA PL/SQL packages that are available for developer users.
- ORDS_METADATA schema
 - ORDS_METADATA schema is granted on the following packages if these packages are not granted EXECUTE on PUBLIC:
 - * SYS.DBMS_ASSERT
 - * SYS.DBMS_LOB
 - * SYS.DBMS_OUTPUT
 - * SYS.DBMS_SCHEDULER
 - * SYS.DBMS_SESSION
 - * SYS.DBMS_UTILITY
 - * SYS.DEFAULT_JOB_CLASS
 - * SYS.HTP
 - * SYS.OWA
 - * SYS.WPG_DOCLOAD
 - ORDS_METADATA is granted SELECT (11g) or READ (12c or later) on the following view if it is not granted SELECT or READ to PUBLIC:
 - * SYS.SESSION_PRIVS
 - ORDS_METADATA schema is granted EXECUTE on the following packages:
 - * SYS.DBMS_CRYPTO
 - * SYS.DBMS_METADATA
 - ORDS_METADATA schema is granted SELECT (11g) or READ (12c or later) on the following views:
 - * SYS.DBA_OBJECTS
 - * SYS.DBA_ROLE_PRIVS
 - * SYS.DBA_TAB_COLUMNS
 - ORDS_METADATA schema is granted SELECT including WITH GRANT OPTION on the following views:
 - * SYS.USER_CONS_COLUMNS
 - * SYS.USER_CONSTRAINTS

- * SYS.USER_OBJECTS
- * SYS.USER_PROCEDURES
- * SYS.USER_TAB_COLUMNS
- * SYS.USER_TABLES
- * SYS.USER_VIEWS
- ORDS_METADATA schema is granted the following system privileges:
 - * ALTER USER
 - * CREATE ANY TRIGGER
 - * CREATE JOB
 - * CREATE VIEW
 - * CREATE PUBLIC SYNONYM
 - * DROP PUBLIC SYNONYM
- ORDS_METADATA schema is granted the necessary object privileges to migrate Application Express REST data to ORDS_METADATA tables.
- ORDS_METADATA schema is granted ORDS_ADMINISTRATOR_ROLE, ORDS_RUNTIME_ROLE roles with administrator option.
- PUBLIC is granted SELECT on many ORDS_METADATA tables and views.
- PUBLIC is granted EXECUTE on PL/SQL packages that are available for users to invoke.
- ORDS_METADATA is granted EXECUTE on the following packages if these packages are not granted EXECUTE to PUBLIC:
 - SYS.DBMS_ASSERT
 - SYS.DBMS_LOB
 - SYS.DBMS_OUTPUT
 - SYS.DBMS_SCHEDULER
 - SYS.DBMS_SESSION
 - SYS.DBMS_UTILITY
 - SYS.DEFAULT_JOB_CLASS
 - SYS.HTP
 - SYS.OWA
 - SYS.WPG_DOCLOAD
- ORDS_METADATA is granted the necessary object privileges to migrate Application Express REST data to ORDS_METADATA tables.

1.6 About the Database Users Used by Oracle REST Data Services

Oracle REST Data Services uses the following database users:

User Name	Required	Description
ORDS_METADATA	Yes	Owner of the PL/SQL packages used for implementing many Oracle REST Data Services capabilities. ORDS_METADATA is where the metadata about Oracle REST Data Services-enabled schemas is stored. It is not accessed directly by Oracle REST Data Services; the Oracle REST Data Services application never creates a connection to the ORDS_METADATA schema.
ORDS_PUBLIC_USER	Yes	User for Oracle REST Data Services to make an initial database connection for handling requests. Depending on the request, the user will proxy to the relevant Oracle REST Data Services-enabled schema or PL/SQL Gateway user to complete the requested action in the database.

1.7 About Command-Line Interface

ORDS release 22.1.0 introduces the ORDS Command-Line Interface that provides the following benefits:

- Create, update, and list your ORDS configuration
- Add additional database pools to your configuration Install/upgrade
- Repair, or uninstall ORDS in the database
- Run ORDS in standalone mode

The preceding functions can be accomplished interactively through prompts, or run silently (non-interactively) using the ORDS commands.

1.8 About Online Help

The online help provides information about the commands along with the available options and arguments. To show the list of ORDS commands, execute the following command:

```
ords --help
```

To show the details of an ORDS command, specify the ORDS command followed by `--help`. If you want additional details for a sub-command, then specify the `--help` after the sub-command.

Syntax:

```
ords <command> --help
```

```
ords <command> <sub-command> --help
```

Examples:

- `ords install --help`
- `ords config --help`

`ords config set --help`

2

Installing and Configuring Oracle REST Data Services

This chapter describes how to install, configure, and upgrade Oracle REST Data Services (ORDS).

Topics:

- [Installing Oracle REST Data Services](#)
- [Repairing the Oracle REST Data Services Installation](#)
- [Upgrading Oracle REST Data Services](#)
- [Uninstalling Oracle REST Data Services](#)
- [Updating the Configuration Settings](#)

Related Topics

- [Using the Multitenant Architecture with Oracle REST Data Services](#)
- [About the Database Users Used by Oracle REST Data Services](#)

2.1 Installing Oracle REST Data Services

You must complete the following steps before installing ORDS.

Note:

The procedures in this topic apply to installing Oracle REST Data Services in a single instance Database or a PDB. For setting up ORDS in a CDB environment, refer to [Using the Multitenant Architecture with Oracle REST Data Services](#) section.

- [Downloading ORDS](#)
- [Setting Up the Configuration Folder Structure](#)

You can then install ORDS using one of the following options:

- [Interactive Command-Line Interface Installation](#)
- [Non-Interactive Command-Line Interface Installation \(Silent\)](#)

See Also:

- [Oracle REST Data Services Downloads](#)
- [Oracle REST Data Services SODA for REST Developer's Guide](#)

2.1.1 Downloading ORDS

This section describes the how to download ORDS.

Downloading ORDS Zip file

Download the `ords_<latest>.zip` file from the Oracle REST Data Services (ORDS) download page, and then unzip `ords_<latest>.zip` file into a folder of your choice. The folder you choose to unzip the file is referred to as the ORDS product folder. The ORDS product folder contains a `bin` folder and other folders and files required to run ORDS.

Adding ORDS `bin` folder to your Operating System PATH

Oracle recommends to add the ORDS `bin` folder to your operating system PATH environment variable.

Example for LINUX/UNIX Operating System

```
echo -e 'export PATH="$PATH:<ords product folder>/bin"' >> ~/.bash_profile
```

Start a new shell to pick up this change.

Example for WINDOWS Operating System

Use a command prompt with administrator privileges.

```
SetX PATH "%PATH%;<ords product folder>\bin"
```

Start a new command prompt to pick up this change.

Example for MAC Operating System

```
echo -e 'export PATH="$PATH:<ords product folder>/bin"' >> ~/.zprofile
```

Start a new terminal to pick up this change.

Downloading ORDS Using yum

ORDS is published in OL7 and OL8 repositories. The system administrator can then add this repository to the yum configuration in order to install the ORDS and handle its dependencies.

OL7 repo: https://yum.oracle.com/repo/OracleLinux/OL7/oracle/software/x86_64/

OL8 repo: https://yum.oracle.com/repo/OracleLinux/OL8/oracle/software/x86_64/

If the oracle software repository has been added to your yum configuration, then you can download and install ORDS using the following command:

```
sudo yum install ords
```

The preceding command handles all the ORDS dependencies and place the ORDS program in `/usr/local/bin/ords` folder to set up your ORDS configuration and install or upgrade ORDS in the database.

Oracle recommends you to use `/etc/ords/conf` configuration directory. The following command is used to configure ORDS:

```
ords --config /etc/ords/config install
```

If you want to use a different directory to place your ORDS configurations and you want to use the Linux system service management as `systemd` or `system v`, then configure the ORDS config directory in `/etc/ords.conf` folder.

Installing ORDS Using RPM

If your system where you want to install is not using `yum` or is not connected to the internet, then you can download the ORDS RPM from the links mentioned in the preceding section (Download ORDS using `yum`) and install the RPM manually. This does not handle the ORDS dependencies, you must install the `java JDK` manually. After downloading the RPM file you can install it using the following command:

```
sudo rpm -i ords-22.1.X-X.elX.noarch.rpm
```

This places the ORDS program in `/usr/local/bin/ords` folder to set up your ORDS configuration and install or upgrade ORDS in the database.

To configure your ORDS, we recommend using the configuration directory in `/etc/ords/conf` folder with the following command:

```
ords --config /etc/ords/config install
```

If you want to use a different directory to place your ORDS configurations and you want to use the LINUX system service management as `systemd` or `system v`, then configure the ORDS configuration directory in `/etc/ords.conf` folder.

2.1.2 Setting Up the Configuration Folder Structure

This section describes how to set up the configuration folder structure.

2.1.2.1 Specifying the New ORDS Configuration Folder

New installation has the ORDS configuration files placed in the configuration folder. Configuration files from ORDS release 21.4.x or earlier are migrated to the configuration folder.

You can obtain the configuration folder location using one of the following options:

1. Command option:

Use the `--config` option to specify your configuration folder.

Example:

```
ords --config /path/to/conf install
```

2. Environment variable:

 **Note:**

This option is preferred if you forget to include the `--config` option when you are using the ORDS command-line Interface.

- **ORDS_CONFIG: Create the ORDS_CONFIG environment variable.**
Example for LINUX or UNIX operating system:

```
Shell script containing ORDS_CONFIG
cat example_env
export ORDS_CONFIG=/path/to/conf
```

```
echo $ORDS_CONFIG
/path/to/conf
```

- **JDK_JAVA_OPTIONS:**
Example:

```
export JDK_JAVA_OPTIONS=-Dconfig.url=/Users/<username>/work/dbtools-
dev/config
./ords-22.1.0.087.1756/bin/ords config list
```

```
JDK_JAVA_OPTIONS: -Dconfig.url=/Users/<username>/work/dbtools-dev/
config
```

3. Current Working Directory:

If the `--config <configuration_folder>` option is not specified, and the `ORDS_CONFIG` environment variable is not defined, then your current working directory is used as the configuration folder.

Example:

If your current working directory is `/path/to/conf`, then it uses that location for your configuration folder.

 **Note:**

- Oracle highly recommends you to ensure that ORDS does not create the configuration directory in the ORDS product folder. For example, you can have a configuration folder in the following location:

```
/Users/<user_name>/work/dbtools-dev/config/
```

The best practice is to have your configuration files separate from the application files, this makes maintenance and upgrades easier and more reliable.

- The following command is no longer valid:

```
Example: java -jar ords.war install
```

If you specify the legacy commands using `java -jar ords.war`, you get the following warning message:

Warning: Support for executing: `java -jar ords.war` has been deprecated. Please add `ords` to your `PATH` and use the `ords` command instead. Run the following command to add ORDS to your `PATH`:

```
<Displays an example of adding the bin folder to your PATH>
```

Start a new terminal to pick up this change. Oracle recommends to add the ORDS product `bin` folder to your path.

2.1.3 Interactive Command-Line Interface Installation

This section explains the interactive command-line interface (CLI) installation prompts. They prompt you for necessary information to create or update the ORDS configuration files, install or upgrade ORDS schema, ORDS database user, and related database objects in the database. You have the option to run in standalone mode.

To be prompted, you must execute one of the following commands:

- Specify the install command. For example: `$ ords install`

 **Note:**

Assumes that the configuration directory is specified through the environment variable or you are defaulting to the current working directory.

- Specify the option `--config <CONFIG FOLDER>` followed by the install command. For example:
 - `$ ords --config /path/to/config install`
 - `ords --config /path/to/conf install --interactive --log-folder /path/to/logs`

Table 2-1 Interactive Installation Prompts

Pro mpt Num ber	Prompt	Description
1.	Enter a number to select the type of installation [1] Install or upgrade ORDS in the database only [2] Create or update a database pool and install/upgrade ORDS in the database [3] Create or update a database pool only Choose [2]:	Refer to Entering a Number to Select the Type of Installation .
2.	Enter a number to select the database pool to update, or create an additional database pool. The selected (or created) database pool will be used to install or upgrade ORDS in the database. [1] default jdbc:oracle:thin:@//localhost:1521/orcl [2] Create an additional database pool Choose [1]:	Refer to Entering a Number to Select the Database Pool
3.	Enter the database pool name:	Refer to Entering the Database Pool Name
4.	Enter a number to select the database connection type to use [1] Basic (host name, port, service name) [2] TNS (TNS alias, TNS directory) [3] Custom database URL Choose [1]:	Refer to Entering a Number to Select the Database Connection
5.	Enter the database host name [localhost]:	Refer to Entering the Database Host Name
6.	Enter the database listen port [1521]:	Refer to Entering the Database Listener Port
7.	Enter the database service name [orcl]:	Refer to Entering the Database Service Name

Table 2-1 (Cont.) Interactive Installation Prompts

Prompt Number	Prompt	Description
8.	Enter the TNS location: /path/to/tns	Refer to Entering the TNS Location
9.	Enter a number to select the TNS Network alias to use [1] DB1 ...CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=pdb1) [2] DB2 ...CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=pdb2)) [3] DB3 ...CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=pdb3)) Choose [1]:	Refer to Entering a Number to Select the TNS Network Alias
10.	Enter the Custom database URL:	Refer to Entering the Custom Database URL
11.	Provide database username with administrator privileges. Enter the administrator username: SYS	Refer to Entering the Administrator Username
12.	Enter the database password for SYS AS SYSDBA:	Refer to Entering the Database Password
13.	Enter the default tablespace for ORDS_METADATA and ORDS_PUBLIC_USER [SYSAUX]: Enter the temporary tablespace for ORDS_METADATA and ORDS_PUBLIC_USER [TEMP]:	Refer to Entering the Default Tablespace
14.	Enter a number to select additional feature(s) to enable: [1] Database Actions (Enables all features) [2] REST Enabled SQL and Database API [3] REST Enabled SQL [4] Database API [5] None Choose [1]:	Refer to Entering a Number to Select Additional Feature(s)

Table 2-1 (Cont.) Interactive Installation Prompts

Pro mpt Num ber	Prompt	Description
15.	Enter a number to configure and start ORDS in standalone mode [1] Configure and start ORDS in standalone mode [2] Skip Choose [1]:	Refer to Entering a Number to Configure and Start ORDS in Standalone Mode
16.	Enter a number to use HTTP or HTTPS protocol [1] HTTP [2] HTTPS Choose [1]:	Refer to Entering a Number to Use HTTP or HTTPS Protocol
17.	Enter the HTTP port [8080]:	Refer to Entering the HTTP Port
18.	Enter the HTTPS port [8443]:	Refer to Entering the HTTPS Port
19.	Enter a number to select the certificate type [1] Use self-signed certificate (generates automatically) [2] Use my SSL certificate (requires SSL certificate and SSL certificate private key) Choose [1]:	Refer to Entering a Number to Select the Certificate Type
20.	Enter the SSL hostname:	Refer to Entering the SSL Hostname
21.	Enter the path for the SSL Certificate:	Refer to Entering the Path for the SSL Certificate
22.	Enter the path for the SSL Certificate's private key:	Refer to Entering the Path for the SSL Certificate's Private Key

Table 2-1 (Cont.) Interactive Installation Prompts

Pro mpt Num ber	Prompt	Description
23.	Enter the APEX static resources location: / path/to/apex/images	Refer to Entering the APEX Static Resources Location

2.1.3.1 Entering a Number to Select the Type of Installation

Select the type of installation.

1. Option 1

- Install or upgrade ORDS in the database only
 - Installs the ORDS schema, ORDS database user, and related database objects if ORDS does not exist in the database.
 - Upgrades ORDS if the schema version is less than the product version.

2. Option 2

- Create or update a database pool and install/upgrade ORDS in the database
 - Create a database pool, or select an existing database pool to update.
 - For an existing database pool, when prompted, it displays the setting value as the default values.
- Install/Upgrade ORDS in the database
 - Installs the ORDS schema, ORDS database user, and related database objects if ORDS does not exist.
 - Upgrades ORDS if the schema version is less than the product version.

3. Option 3

- Create or update a database pool only
 - Create a database pool, or select an existing database pool to update.
 - For an existing database pool, when prompted, it displays the setting value as the default values.

2.1.3.2 Entering a Number to Select the Database Pool

If the ORDS database pool(s) already exist in your configuration folder, it will display the database pool(s) along with its connection.

You can select the database pool, or

Create an additional database pool if you selected

- option 2 for prompt 1
- option 3 for prompt 1

 **Note:**

If the ORDS configuration pool does not exist, then it prompts you for the database connection type. See prompt 3.

2.1.3.3 Entering the Database Pool Name

If you select the option 2 for prompt number 2, then the database pool name prompt is displayed. Enter the database pool name.

 **Note:**

- The database pool name must only contain lowercase alphabets a-z, digits 0-9, and the "-" character and must start with a lowercase alphanumeric character and end with a lowercase alphanumeric character.
For example, if the pool name you enter is `database1`, then all REST requests to that pool will start with `/ords/database1/`.
- When the database pool is created, the database pool folder is also created along with the pool specific settings, and a wallet to store the database user credentials.
- This prompt also sets the mapping pattern. See [Configuring Additional Databases](#) for more information.

2.1.3.4 Entering a Number to Select the Database Connection

Select the database connection type.

 **Note:**

If ORDS configuration pool does not exist in your configuration folder, then after selecting from prompt number 1, prompt number 4 is displayed.

If option 2 or option 3 is selected for prompt 1, then the database pool folder named `default` is created along with the settings specific to the pool, and a wallet to store the database user credentials.

2.1.3.5 Entering the Database Host Name

Specify the database host name for basic connection.

2.1.3.6 Entering the Database Listener Port

Specify the database port for a basic connection.

2.1.3.7 Entering the Database Service Name

Specify the service name. The default value for the service name is obtained from one of the following in the given sequence:

1. The service name setting in the ORDS configuration file. If not present, then
2. It checks for `ORACLE_PDB` environment variable. If undefined, then
3. It checks for `ORACLE_SID` environment variable. If undefined, then
4. It uses `orcl` as the default value.

2.1.3.8 Entering the TNS Location

Specify the location for `tnsnames.ora` file. The location can be to the folder where `tnsnames.ora` file is located, or a zip file that contains the `tnsnames.ora` file for TNS connection.

2.1.3.9 Entering a Number to Select the TNS Network Alias

Select the TNS alias name from the list of TNS network alias names displayed.

2.1.3.10 Entering the Custom Database URL

Specify the custom database URL for custom database URL connection.

2.1.3.11 Entering the Administrator Username

Specify a user who has installer privileges to install or upgrade ORDS in the database.



See Also:

[Privileges Granted to the Oracle REST Data Services Users](#)

2.1.3.12 Entering the Database Password

Specify the administrator user password.

2.1.3.13 Entering the Default Tablespace

Specify the tablespaces for both the ORDS schema and ORDS database user.



Note:

If ORDS is not installed in the database, then you are prompted for the tablespaces.

2.1.3.14 Entering a Number to Select Additional Feature(s)

Select the additional features that you want to enable.

2.1.3.15 Entering a Number to Configure and Start ORDS in Standalone Mode

You can configure ORDS to run in standalone mode. In addition, you can start ORDS in standalone mode after the install or upgrade is complete.

If you do not want to configure ORDS and start in standalone mode, choose Option 2 to exit.

2.1.3.16 Entering a Number to Use HTTP or HTTPS Protocol

Select the protocol (HTTP or HTTPS) that you want to use.



Note:

It is highly recommended to run ORDS in Standalone mode with HTTPS versus HTTP.

2.1.3.17 Entering the HTTP Port

Specify the HTTP port, if you selected option 1: HTTP for prompt number 16.

2.1.3.18 Entering the HTTPS Port

Specify the HTTPS port, if you selected option 2: HTTPS for prompt number 16.

2.1.3.19 Entering a Number to Select the Certificate Type

Select an option for the certificate type, if you selected option 2: HTTPS Protocol for prompt number 16.

If option 1 is selected, then ORDS automatically generates the self-signed certificate and prompts you for the SSL hostname.

If option 2 is selected, then you get additional prompts for the certificate path and certificate private key path. Refer to SSL Certificate section for more details.

2.1.3.20 Entering the SSL Hostname

Specify the SSL hostname, if you selected option 1: Use self-signed certificate for prompt 19.

2.1.3.21 Entering the Path for the SSL Certificate

Specify the SSL certificate path, if you selected option 2: Use my SSL certificate for prompt number 19.

2.1.3.22 Entering the Path for the SSL Certificate's Private Key

Specify the path for the SSL Certificates' private key, if you have selected option 2: Use my SSL certificate for prompt 19.

2.1.3.23 Entering the APEX Static Resources Location

ORDS detects if APEX is installed in the database and if the database user `APEX_PUBLIC_USER` exists, if found to be true, then it prompts for the location of the APEX images.

2.1.4 Non-Interactive Command-Line Interface Installation (Silent)

The non-interactive command-line interface (CLI) installation or silent installation, performs the following tasks:

- Creates or updates the ORDS configuration files
- Installs or upgrades the ORDS schema
- Creates the ORDS proxy user and the related objects in the Database

The non-interactive installation command does not run in standalone mode. If you want to run it in standalone mode, then it must be followed by the `serve` command.

If you provide the command-line options for the non-interactive installation and if you are missing a required option (for example: Basic connection requires the `db-port` option), then an error message is displayed.

The `--db-pool <pool_name>` option specifies the database pool to create or update. If the `--db-pool` option is omitted, then it defaults to the `default` database pool.

2.1.4.1 Understanding Command Options for Command-Line Interface Installation

Table 2-2 Command Options for Command-Line Interface Installation

Options	Description
<code>--admin-user <USERNAME></code>	Specifies the user with privileges to install or upgrade ORDS in the database, or obtains the ORDS status information.
<code>--bequeath-connect</code>	Specifies that bequeath connect that enables clients to connect directly to an Oracle database bypassing the network listener. This is when the client is on the same server as the database. Refer to Oracle REST Data Services documentation to setup your environment to use the <code>--bequeath-connect</code> option.
<code>--config <CONFIG_PATH></code>	Specifies the path to the folder containing the configuration files.
<code>--config-only</code>	Specifies to create or update the configuration only.
<code>--db-only</code>	Specifies to install or upgrade ORDS in the database and not to create or update the configuration files.
<code>--db-pool <POOL_NAME></code>	Specifies the name of the database pool to create, update or use. If the <code>db-pool</code> option is omitted, then the <code>default</code> database pool is used.
<code>--db-custom-url <URL></code>	Specifies the custom database url for the database connection.
<code>--db-hostname <HOST></code>	Specifies the database host name.
<code>--db-port <PORT></code>	Specifies the database port.
<code>--db-servicename <SERVICENAME></code>	Specifies the database service name.
<code>--db-sid <SID></code>	Specifies the database SID.

Table 2-2 (Cont.) Command Options for Command-Line Interface Installation

Options	Description
<code>--db-tns-alias <ALIAS_NAME></code>	Specifies the TNS alias name that exists in the <code>tnsnames.ora</code> file.
<code>--db-tns-dir <TNS_DIR></code>	Specifies the folder where the <code>tnsnames.ora</code> file is located.
<code>--feature-db-api <BOOLEAN></code>	Specifies if you want to enable DB API feature. Possible values are: <code>true</code> or <code>false</code> . <ul style="list-style-type: none"> If the value is set to <code>true</code>, then DB API feature is enabled. If the value is set to <code>false</code>, then DB API feature is disabled. Returns an error if the specified options are <code>--feature-sdw true</code> and <code>--feature db-api false</code> .
<code>--feature-rest-enabled-sql <BOOLEAN></code>	Specifies if you want to enable REST-Enabled SQL feature. Possible values are: <code>true</code> or <code>false</code> . <ul style="list-style-type: none"> If the value is set to <code>true</code>, then the REST-Enabled SQL feature is enabled. If the value is set to <code>false</code>, then the REST-Enabled SQL feature is disabled. Returns an error if the specified options are <code>--feature-sdw true</code> and <code>--feature-rest-enabled-sql false</code> .
<code>--feature-sdw <BOOLEAN></code>	Specifies if you want to enable Database Actions feature. Possible values are: <code>true</code> or <code>false</code> . <ul style="list-style-type: none"> If the value is set to <code>true</code>, then the Database Actions feature is enabled. If the value is set to <code>false</code>, then the Database Actions feature is disabled. If the option is set to <code>true</code> , then the following settings are set to <code>true</code> in the configuration file: <ul style="list-style-type: none"> <code>database.api.enabled</code> <code>restEnabledSql.active</code> Returns an error if <code>--feature-sdw true</code> and any of following options are specified, and are set to <code>false</code> : <ul style="list-style-type: none"> <code>--feature-db-api</code> <code>--feature-rest-enabled-sql</code>
<code>--gateway-mode <MODE></code>	Specifies the PL/SQL gateway mode. Possible values are: <code>proxied</code> , <code>direct</code> , or <code>disabled</code> . Default value is <code>disabled</code> . Where: <ul style="list-style-type: none"> <code>disabled</code>- PL/SQL gateway is not allowed. <code>direct</code> - The PL/SQL gateway call is performed directly as the user identified by <code>db.username</code> (located in the <code>pool.xml</code> setting). <code>proxied</code> - The PL/SQL gateway call is performed by proxying from the run-time user (<code>db.username</code> is located in the <code>pool.xml</code>) to the relevant schema configured by <code>ords_admin.config_plsql_gateway</code>.
<code>--gateway-user <USER></code>	Specifies the user that exists in the Oracle database and has privileges to access the stored procedures.

Table 2-2 (Cont.) Command Options for Command-Line Interface Installation

Options	Description
<code>--interactive, -i</code>	Prompts user for all the required information.
<code>--help, -h</code>	Shows how to use the command.
<code>--legacy-config <FOLDER></code>	Specify the legacy configuration folder to migrate to the revised configuration structure.
<code>--legacy-context <STRING></code>	Specifies the context (example: <code>ords</code> , <code>apex</code> , and so on.) that was used in the legacy configuration.
<code>--log-folder</code>	Writes the logs from the install, upgrade, repair or uninstall to the log folder. If this option is omitted, then the output is written to standard output.
<code>--password-stdin</code>	To run the ORDS install, upgrade, repair, or uninstall command non-interactively, use this option to read the password value from standard input when redirecting input to a file or here document.
<code>--pdb-exclude <(PDB...)></code>	Specifies the option applicable for CDB. List of PDBs that are excluded from the ORDS install/upgrade, or repair.
<code>--pdb-open-readwrite-all</code>	Specifies the option applicable for CDB. Specifies whether to open all PDBs in read write mode if their status is either closed or read only.
<code>--pdb-open-readwrite <(PDB...)></code>	Specifies the option applicable for CDB. List of PDBs that are read-only or closed to be open for read/write to install, upgrade, repair, or uninstall ORDS.
<code>--pdb-skip-readonly</code>	Specifies the option applicable for CDB. Skips PDBs that are read-only. If this option is omitted, an error message is returned informing the user that the PDB(s) are read-only for install or upgrade ORDS. Excludes <code>PDB\$SEED</code> since it is open for read/write.
<code>--pdb-skip-closed</code>	Specifies the option applicable for CDB. Skips PDBs that are closed (mounted). If this option is omitted, an error message is returned informing the user that the PDB(s) are closed for installation/upgradation of ORDS.
<code>--proxy-user</code>	Specifies the password of the ORDS database user (<code>ORDS_PUBLIC_USER</code>) that is in a file or in a here document when redirecting the standard input.
<code>--proxy-user-tablespace <TABLESPACE></code>	Specifies the option applicable for ORDS install. The default tablespace for proxy user (For example: <code>ORDS_PUBLIC_USER</code>). If this option is omitted, then the defaulted database default tablespace is used.
<code>--proxy-user-temp-tablespace <TABLESPACE></code>	Specifies the option applicable for an ORDS install. The temporary tablespace for proxy user (For example: <code>ORDS_PUBLIC_USER</code>). If this option is omitted, then the defaulted database temporary tablespace is used.

Table 2-2 (Cont.) Command Options for Command-Line Interface Installation

Options	Description
<code>--schema-tablespace <TABLESPACE></code>	Specifies the default tablespace for ORDS_METADATA schema. If this option is omitted, then the defaulted database default tablespace is used.
<code>--schema-temp-tablespace <TABLESPACE></code>	Specifies the option applicable for an ORDS install. The temporary tablespace for ORDS_METADATA schema. If this option is omitted, then the defaulted database temporary tablespace is used.

2.1.4.2 Using Input Redirection

This section describes how to redirect the standard input using the here command or to a file.

Redirect STDIN to a file

Redirect STDIN to a file that contains the password. In the following example, the file must contain two passwords. Each password must be on a separate line.

Example:

```
$ cat password.txt
firstpassword
secondpassword
```

The first password belongs to the first user on the command line (for example: `SYS`). The second password belongs to the second user on the command line (for example: `proxy-user`).

```
ords --config <configuration_folder> install --db-pool <pool_name> --admin-
user <username> --proxy-user
--db-hostname <host> --db-port <port_number> --db-servicename <service_name>
--log-folder <log_folder> --feature-sdw <boolean> --password-stdin <
<filename>
```

Example:

```
ords --config /path/to/conf install --db-pool db1 --admin-user SYS --proxy-
user --db-hostname localhost
--db-port 1521 --db-servicename orcl --log-folder /path/to/logs --feature-sdw
true
--password-stdin < password.txt
```

Redirect Standard Input Using Here Document

Redirect STDIN using the Here document (also known as heredoc) for the password(s). The heredoc consists of the `<<<` redirection operator followed by a delimiter token.

Each password must be on a separate line and it is ended by the delimiter token.

Example:

The first password belongs to the first user on the command line (for example: `SYS`). The second password belongs to the second user on the command line (for example: `proxy-user`).

```
ords --config <configuration_folder> install --db-pool <pool_name> --admin-
user <username> --proxy-user --db-hostname
<host> --db-port <port_number> --db-servicename <service_name> --log-folder
<log_folder> --feature-sdw <boolean>
--password-stdin << EOF
<password1>
<password2>
EOF
```

```
ords --config /path/to/conf install --db-pool db1 --admin-user SYS --proxy-
user --db-hostname
localhost --db-port 1521 --db-servicename orcl --log-folder /path/to/logs --
feature-sdw true
--password-stdin << EOF
<password1>
<password2>
EOF
```

2.2 Repairing the Oracle REST Data Services Installation

If you want to check that the Oracle REST Data Services installation is valid, specify the repair command.

**Note:**

To repair ORDS in the database, the ORDS product version must be the same version as the ORDS installed schema version in the database.

Example of repair command to prompt for information:

```
ords [--config <folder>] install repair --interactive [--log-folder <folder>]
```

If you have an existing configuration, you may choose to select from a list of database pools. You also have the option to specify the database connection for ORDS to be installed. You are prompted to provide the administrator username and password.

Example

Oracle REST Data Services - Interactive Repair

```
Enter a number to select the database pool to use or specify the database
connection
```

```
[1] default          jdbc:oracle:thin:@//localhost:1521/orcl
[2] sales            jdbc:oracle:thin:@//localhost:1521/pdb1
[3] Specify the database connection
```

```
Choose [1]:
Provide database user name with administrator privileges.
Enter the administrator username: SYS
Enter the database password for SYS AS SYSDBA:
```

 **Note:**

When you install Oracle REST Data Services, it attempts to find the Oracle APEX schema and creates a view. This view joins the relevant tables in the APEX schema to the tables in the Oracle REST Data Services schema. If you install Oracle REST Data Services before APEX, then Oracle REST Data Services cannot find the APEX schema and it creates a stub view in place of the missing APEX tables.

Oracle highly recommends that you install Oracle REST Data Services after APEX to ensure that the APEX objects, which Oracle REST Data Services needs to query, are present. If you install Oracle REST Data Services before APEX, then use the repair command to force Oracle REST Data Services to reconstruct the queries against the APEX schema.

Run the following command to ensure that the ORDS configuration for your database pool contains the PL/SQL gateway mode setting and is set to proxied:

```
ords --config <config_path> --db-pool <pool_name> get
plsql.gateway.mode
```

Run the following command to set `plsql.gateway.mode` value to proxied :

```
ords --config <config_path> config --db-pool <pool_name> set
plsql.gateway.mode proxied
```

2.2.1 Command Options for Repair CLI

This section describes the interactive and non-interactive install repair CLI commands used to repair the ORDS schema in the database.

Table 2-3 Command Options for Repair CLI

Command	Description
<code>--admin-user <string></code>	Specifies the user with privileges to install, upgrade, or uninstall ORDS in the database.
<code>--bequeath-connect</code>	Specifies that the bequeath connect enables clients to connect directly to an Oracle database bypassing the network listener. This occurs if client is on the same server as the database.
<code>--db-custom-url <url></code>	Specifies the custom database url for the database connection.
<code>--db-hostname <string></code>	Specifies the database host name.
<code>--db-pool <string></code>	Specifies the name of the database connection pool.
<code>--db-port <int></code>	Specifies the database port.

Table 2-3 (Cont.) Command Options for Repair CLI

Command	Description
<code>--db-servicename <string></code>	Specifies the database service name.
<code>--db-sid <string></code>	Specifies the system identifier which is a unique name that identifies the database.
<code>--db-tns-alias <string></code>	Specifies the TNS alias name that exist in the <code>tnsnames.ora</code> file.
<code>--db-tns-dir <folder></code>	Specifies the folder where the <code>tnsnames.ora</code> file is located.
<code>-h, --help</code>	Shows how to use and provides information of the command.
<code>-i, --interactive</code>	Prompts for the required information.
<code>--log-folder <folder></code>	Writes the logs from the install, upgrade or uninstall to the log folder. If this option is omitted, then the output is written to standard output.
<code>--password-stdin</code>	Specifies to use this option to read the password value from standard input when redirecting input to a file or here document when you run the <code>ords install</code> or <code>ords uninstall</code> command non-interactively, If the <code>--password-stdin</code> option and the redirection symbol (" <code><</code> " or " <code><<</code> ") are omitted, then you are prompted for the password(s) for the command option(s) that require the password.
<code>--pdb-exclude <string...></code>	Specifies the list of PDBs that are excluded from the ORDS install, upgrade, or repair. This option only applies to the CDB.
<code>--pdb-open-readwrite <string...></code>	Specifies the list of PDBs that are read only or closed to be open for read write for ORDS install, upgrade, repair or uninstall. This option only applies to the CDB.
<code>--pdb-open-readwrite-all</code>	Specifies to open all PDBs in read write mode if their status is either closed or read only. This option only applies to the CDB.
<code>--pdb-skip-closed</code>	Skips PDBs that are closed (mounted). This option only applies to the CDB.
<code>--pdb-skip-readonly</code>	Skips PDBs that are read-only. This option only applies to the CDB.

2.3 Upgrading Oracle REST Data Services

This section describes how to upgrade to Oracle REST Data Services latest release.

Upgrading from ORDS 22.1.x or Later

If you are upgrading from ORDS 22.1.x or later releases, then refer to the following sections:

- [Interactive Command-Line Interface Installation](#)
- [Non-Interactive Command-Line Interface Installation \(Silent\)](#)

Upgrading from ORDS 21.4.x or earlier releases

If you are upgrading from ORDS 21.4.x or earlier releases, then refer to [Migrate Configuration and Upgrade from ORDS 21.4.x or Earlier Releases](#)

2.3.1 Migrate Configuration and Upgrade from ORDS 21.4.x or Earlier Releases

This section describes how to upgrade from ORDS 21.4.x or earlier releases.

To upgrade to ORDS 22.x from ORDS 21.4.x or earlier releases:

Provide the location of your configuration files that were used from your previous ORDS release (21.4.x or earlier). If you do not know the location, then specify the command using your previous ORDS release (21.4.x or earlier).

```
java -jar /path/to/earlierRelease/ords.war configdir
```

Example:

```
java -jar /path/to/earlierRelease/ords.war configdir
```

```
INFO The config.dir value is /path/to/legacy/conf
```

In the preceding example `/path/to/legacy/conf` is configuration location

You can upgrade to ORDS 22.x release from ORDS 21.4.x or earlier releases in the following two ways:

- [Interactive Upgrade](#)
- [Silent Upgrade](#)

2.3.1.1 Interactive Upgrade

Interactive Upgrade

Specify the following install command to migrate your configuration, and to upgrade ORDS in your specified database:

```
ords --config <folder> install -i --legacy-config <folder> --log-folder  
<folder>
```

Table 2-4 Interactive Upgrade Commands

Command Options	Description
<code>--config <folder></code>	Specify the configuration folder to store your migrated configuration files. Your configuration folder must not be in the same location as your legacy configuration folder. Otherwise, an error occurs.
<code>-i</code> or <code>--interactive</code>	Specifies to run the command in interactive mode. You are then prompted for the information.
<code>--legacy-config <folder></code>	Specifies to provide the location of your configuration files that you used for ORDS 21.4.x or earlier releases.
<code>--log-folder <folder></code>	Specifies to provide the location to store your log file.

```
ords --config /path/to/new/conf install -i --legacy-config /path/to/legacy/conf --log-folder /path/to/logs
```

```
ORDS: Release 22.x Production on Wed <month> 1 12:00:01 <year>
```

Copyright (c) 2010, 2022, Oracle.

Configuration:
/path/to/new/conf


```
Oracle REST Data Services 22.x.x.rNNNNNN - Migrate Configuration
Migrating ORDS Configuration files located at /path/to/legacy/conf/ords
. . .
Legacy configuration files located /path/to/legacy/conf/ords are no longer
being used to configure ORDS or its connection pools. Your migrated
configuration files are now located at /path/to/new/conf
```

Oracle REST Data Services - Interactive Install

```
Enter a number to select the type of installation
  [1] Install or upgrade ORDS in the database only
  [2] Create or update a database pool and install/upgrade ORDS in the
database
  [3] Create or update a database pool only
Choose [1]:
```

Select option 1 to upgrade ORDS in your database. A list of the migrated database pools are displayed. The legacy configuration database pool named `apex` is migrated, and the name is changed to `default`.

```
Enter a number to select the database pool to use or specify the database
connection
  [1] default      jdbc:oracle:thin:@//localhost:1521/orcl
  [2] hr          jdbc:oracle:thin:@//localhost:1521/pdb1
  [3] Specify the database connection
Choose [1]:
```

Select the database pool that you want to upgrade. Then you are prompted for the username and password of the administrator.

```
Provide database user name with administrator privileges.
  Enter the administrator username: sys
  Enter the database password for SYS AS SYSDBA:
Connecting to database user: SYS AS SYSDBA url: jdbc:oracle:thin:@//
localhost:1521/orcl
```

```
Retrieving information.
Connecting to database user: ORDS_PUBLIC_USER url:jdbc:oracle:thin:@//
localhost:1521/orcl
Created folder /path/to/logs/
INFO   Upgrading Oracle REST Data Services schema 21.4.2.rNNNNNN to version
22.x.x.rNNNNNN in ORCL
INFO   Completed upgrade for Oracle REST Data Services version
22.x.x.rNNNNNN. Elapsed time: 00:00:02.585
```

```
INFO   Log file written to /path/to/logs/ords_upgrade_<date-time>.log
```

This completes the upgradation process.

If you have multiple database pools to upgrade, then perform the install command again excluding the `--legacy-config` option since you have already migrated the configuration.

```
ords --config /path/to/new/conf install -i --log-folder /path/to/logs
```

```
. . .
Enter a number to select the type of installation
  [1] Install or upgrade ORDS in the database only
  [2] Create or update a database pool and install/upgrade ORDS in the
database
  [3] Create or update a database pool only
Choose [1]:
```

Select option 1

2.3.1.2 Silent Upgrade

To do a silent upgrade you must provide the following:

- Legacy configuration location
- The database password for the administrator user

The database password can be in a text file or using a `Here` document. If you are using a text file, then ensure to delete the text file when the upgradation is complete. The legacy configuration database pool named `apex` is migrated, and the name is changed to `default`.

If you omit the `--database-pool <name>` option, then it uses the default database pool to upgrade.

Using a Text File

You can redirect standard input using a text file that contains the password. Delete the text file once you complete the upgrade operation.

```
ords --config <folder> install --admin-user <user> --legacy-config <folder> --
log-folder <folder>
      --password-stdin < <text-file>
```

Example of using a `<text-file>` with 1 password

The text file contains 1 password for `--admin-user <user>`

```
ords --config /path/to/new/conf install --admin-user SYS --legacy-config /
path/to/legacy/conf
--log-folder /path/to/logs --password-stdin < adminpwd.txt
```

Using a Here Document

You can redirect Standard Input using the `Here` document (also known as `heredoc`) for the password(s). The `heredoc` consists of the `<<` redirection operator followed by a delimiter token once the upgrade operation completes. If you are using a script that contains the password(s), then delete the script.

```
ords --config <folder> install --admin-user <user> --legacy-config <folder> --
log-folder <folder> --password-stdin << <delimiter-token>
```

```
> <password>
> <delimiter-token>

ords --config /path/to/new/conf install --admin-user SYS --legacy-config /
path/to/legacy/conf --log-folder
/path/to/logs --password-stdin << EOF
> <password for admin-user>
> EOF
```

2.3.1.3 Upgrading the Multiple Database Pools

Oracle highly recommends that if you have multiple database pools that uses a PL/SQL gateway user, then you must also upgrade those database pools using the `ords install` command so that the PL/SQL gateway user is configured in the ORDS schema.

2.4 Uninstalling Oracle REST Data Services

This section describes the interactive and non-interactive uninstallation CLI commands used to uninstall the ORDS schema.

2.4.1 Interactive Uninstall CLI

The interactive uninstallation CLI prompts you for the necessary information to uninstall the ORDS schema, ORDS proxy user and related database objects from the database.

To be prompted, you must execute one of the following commands:

- Specify the uninstall command only. For example: `$ ords uninstall`

Note:

Assumes that the configuration folder is specified through the environment variable or you are defaulting to the current working directory.

- Specify the option `--config <configuration folder>` followed by the uninstall command. For example: `$ ords --config /path/to/config uninstall`
- Specify the `--interactive` option. For example: `$ ords --config /path/to/config uninstall --interactive`

Table 2-5 Interactive Uninstall Prompts

Prompt Number	Prompt	Description
1.	<pre> Enter a number to select the database pool to use or create the database pool [1] default jdbc:oracle:thin:@// localhost:1521/orcl [2] Specify the database connection Choose [1]: </pre>	Refer to Entering a Number to Select the Database Pool
2.	<pre> Enter a number to select the database connection type to use [1] Basic (host name, port, service name) [2] TNS (TNS alias, TNS directory) [3] Custom database URL Choose [1]: </pre>	Refer to Entering a Number to Select the Database Connection Type
3.	<pre> Enter the database host name [localhost]: </pre>	Refer to Entering the Database Host Name
4.	<pre> Enter the database listen port [1521]: </pre>	Refer to Entering the Database Listener Port
5.	<pre> Enter the database service name [orcl]: </pre>	Refer to Entering the Database Service Name
6.	<pre> Enter the TNS location: /path/to/tns </pre>	Refer to Entering the TNS Location
7.	<pre> Enter a number to select the TNS Network alias to use [1] DB1 ...CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=pdb1)) [2] DB2 ...CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=pdb2)) [3] DB3 ...CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=pdb3)) Choose [1]: </pre>	Refer to Entering a Number to Select the TNS Network Alias

Table 2-5 (Cont.) Interactive Uninstall Prompts

Prompt Number	Prompt	Description
8.	Enter the Custom database URL:	Refer to Entering the Custom Database URL
9.	Provide database username with administrator privileges. Enter the administrator username: SYS	Refer to Entering the Administrator Username
10.	Enter the database password for SYS AS SYSDBA:	Refer to Entering the Database Password for SYS AS SYSDBA
11.	ORDS contains 1 enabled/disabled schema: HR Uninstall ORDS in the database [1] Yes [2] No Choose [2]:	Refer to Entering an Option for Uninstalling ORDS

2.4.1.1 Entering a Number to Select the Database Pool

You get this prompt only if the configuration pool(s) exists in the configuration folder.

- If you select a database pool, then it bypasses the database connection type and prompts you for the administrator username and password.
- If you select option 2, then you get the prompt number 2.

Note:

If the configuration pool(s) does not exist in your configuration folder, then it prompts you for the database connection type.

2.4.1.2 Entering a Number to Select the Database Connection Type

Select the database connection type.

2.4.1.3 Entering the Database Host Name

Specify the database host name for basic connection.

2.4.1.4 Entering the Database Listener Port

Specify the database port for a basic connection.

2.4.1.5 Entering the Database Service Name

Specify the service name. The default value for the service name is obtained from one of the following in the given sequence:

1. The service name setting in the ORDS configuration file. If not present, then
2. It checks for `ORACLE_PDB` environment variable. If undefined, then
3. It checks for `ORACLE_SID` environment variable. If undefined, then
4. It uses `orcl` as the default value.

2.4.1.6 Entering the TNS Location

Specify the location for `tnsnames.ora` file. The location can be to the folder where `tnsnames.ora` file is located, or a zip file that contains the `tnsnames.ora` file for TNS connection.

2.4.1.7 Entering a Number to Select the TNS Network Alias

Select the TNS alias name from the list of TNS network alias names displayed.

2.4.1.8 Entering the Custom Database URL

Specify the custom database URL for custom database URL connection.

2.4.1.9 Entering the Administrator Username

Specify a user who has installer privileges to install or upgrade ORDS in the database.



See Also:

[Privileges Granted to the Oracle REST Data Services Users](#)

2.4.1.10 Entering the Database Password for SYS AS SYSDBA

Specify the administrator user password.

2.4.1.11 Entering an Option for Uninstalling ORDS

If the ORDS enabled or disabled schemas exist, and the `--force` command option is not specified, then it displays the enabled or disabled schemas and prompts for a confirmation to uninstall ORDS in the database.

If the `--force` command option is specified, then the uninstall confirmation is not shown,

2.4.2 Non-Interactive Uninstall CLI

The non-interactive uninstall CLI uninstalls the ORDS schema, ORDS proxy user, and related database objects from the database.

Provide the command options for the non-interactive uninstall CLI. If you are missing a required option (for example, `db-port` for a connection), then you are not prompted. Instead, an error is returned.

2.4.2.1 Using Input Redirection

This section describes how to redirect standard input using Here document or to a file.

Redirect Standard Input to a File

Redirect STDIN to a file that contains the password. In the following example, the file contains one password.

Example:

```
$ cat adminpwd.txt
password
```

The password belongs to the administrator user on the command line (for example: `SYS`). This example assumes that the config folder and the `db1` pool already exists.

```
ords --config /path/to/conf uninstall --db-pool db1 --admin-user SYS --log-
folder /path/to/logs --password-stdin < adminpwd.txt
```

Redirect Standard Input using Here Document

Redirect STDIN using the Here document (also known as heredoc) for the password. The heredoc consists of the `<<` redirection operator followed by a delimiter token.

The password must be on a separate line and it is ended by the delimiter token.

Example:

```
ords --config <configuration_folder> uninstall --db-pool <pool_name> --admin-
user <username>
--db-hostname <host> --db-port <port_number> --db-servicename <service_name>
--log-folder <log_folder>
--password-stdin << EOF
<password>
EOF
```

The password belongs to the the administrator user on the command line. (for example: `SYS`).

```
ords --config /path/to/config uninstall --db-pool db1 --admin-user SYS --db-
hostname
localhost --db-port 1521 --db-servicename orcl --log-folder /path/to/
logs << EOF
<password>
EOF
```

2.4.2.2 Command options for Uninstall CLI

Table 2-6 Command Options for Uninstall CLI

Options	Description
<code>--admin-user <USERNAME></code>	Specifies the user with privileges to install, upgrade, repair, or uninstall ORDS in the database.
<code>--bequeath-connect</code>	Specifies that the bequeath connect enables clients to connect directly to an Oracle database bypassing the network listener. This occurs if client is on the same server as the database.
<code>--config <CONFIG_PATH></code>	Specifies the path to the folder containing the configuration files.
<code>--db-pool <POOL_NAME></code>	Specifies the name of the database pool.
<code>--db-custom-url <URL></code>	Specifies the user specified custom url.
<code>-db-hostname <HOST></code>	Specifies the database host name.
<code>--db-port <PORT></code>	Specifies the database port.
<code>--db-servicename <SERVICENAME></code>	Specifies the database service name.
<code>--db-sid <SID></code>	Specifies the database SID.
<code>--db-tns-alias <ALIAS_NAME></code>	Specifies the tns alias name that exists in the <code>tnsnames.ora</code> file.
<code>-db-tns-dir <TNS_DIR></code>	Specifies the folder where the <code>tnsnames.ora</code> file is located.
<code>--force</code>	Specifies that you can uninstall ORDS from the database regardless of the number of ORDS enabled/disabled schemas.
<code>--help, -h</code>	Shows how to use the command.
<code>--interactive, -i</code>	Prompts the user for all the required information.
<code>--log-folder</code>	Writes the logs from the install, upgrade, repair, or uninstall to the log folder. If this option is omitted, then the output is written to the standard output.
<code>-password-stdin</code>	Specifies when you want to run the ORDS install or uninstall command non-interactively, you can use this option to read the password value from standard input when redirecting input to a file or here document. If the <code>--password-stdin</code> option and the redirection symbol (" <code><</code> " or " <code><<</code> ") are omitted, you are prompted for the password(s) for the command option(s) that requires the password.
<code>--pdb-open-readwrite-all</code>	Specifies the option applicable for CDB. Specifies whether to open all PDBs in read write mode if their status is either closed or read only.
<code>--pdb-open-readwrite <(PDB...)></code>	Specifies the option applicable for CDB. Specifies the list of PDBs that are readonly or closed to be open for readwrite to uninstall ORDS.

2.5 Updating the Configuration Settings

The `ORDS config` command is used to update and edit the contents of the global and database pool specific settings in the configuration files.

The `config` command comprises a list of sub-commands. The `config` command has the `--db-pool <POOL_NAME>` option that identifies the database pool to which you want to apply the setting. If this option is omitted, then the default database pool settings get updated.

```
ords config [OPTIONS] [SUB-COMMAND]
```


Where:

- **OPTIONS:**
 - `--db-pool <string>`: The name of the database connection pool to configure
 - `-h, --help`: Show usage information for the command
- **SUB-COMMAND:** `delete`, `get`, `info`, `list`, `secret`, `set`, and `user`

Following is the list of sub-commands along with the descriptions:

info

Shows the description of the specified key, explaining its purpose. Indicates if it is a sensitive value and what is the default value.

Example: `ords config info <KEY>`

set

Sets the `config` key to a value. If you wish the setting to be set at a global level, then use the `--global` option.

Example: `ords config set [--global] <KEY> <VALUE>`

secret

Sets the sensitive values (for example: connection pools password). Use `ords config secret` to set the sensitive values. Some values are sensitive (example: `db.password`). Per Oracle Secure Coding Standards such values must not be passed as command line arguments as this leads to leakage of the value (appears in `ps` output).

Example: `ords config secret --password-stdin <KEY> < <file>`



Note:

If an attempt is made to do `ords config set` on a sensitive value, an error is returned.

Non-Interactive example:

If the `--password-stdin` option is specified, then the value is read from `stdin`.

Example:

```
ords config secret --password-stdin db.password < secret.txt
```

Interactive Example

If `--password-stdin` is not specified, then the user is prompted to enter and confirm the secret value.

Example:

```
ords config secret db.password
```

get

Echo the config value to stdout. If the value is a secret then it's value will be replaced by the text `*****`. If you wish to see the plain text of the secret, then use the `--secret` argument.

Example: `ords config get [--global] [--secret] <KEY>`

list

Show all explicitly configured values for the pool settings and global settings. Secret values are replaced with the text `*****`. Use `--include-defaults` to list all the settings including those with the default values.

Example: `ords config list [--include-defaults]`

The information lists the name of the setting, value, and the source location (global configuration, pool specific configuration, or global/pool wallet). The table includes the columns such as

- **Setting:** Name of the setting
- **Value:** The value of the setting
- **Source:** Location of the global/pool specific setting

For example the output is similar to the following:

Configuration: `/path/to/conf/`

Database pool: `default`

Setting	Value	Source
-----	-----	-----
<code>db.connectionType</code>	<code>basic</code>	Pool
<code>db.hostname</code>	<code>localhost</code>	Pool
<code>db.password</code>	<code>*****</code>	Pool Wallet
<code>db.port</code>	<code>1521</code>	Pool
<code>db.servicename</code>	<code>orcl</code>	Pool
<code>db.username</code>	<code>ORDS_PUBLIC_USER</code>	Pool
<code>feature.sdw</code>	<code>true</code>	Pool
<code>restEnabledSql.active</code>	<code>true</code>	Pool

The column **Source** indicates the origin of the setting value which can be any one of the following:

- **Default:** Uses the default value for the setting
- **Global:** The value is specified in the `global/settings.xml` configuration file
- **Pool:** The value is explicitly configured in the pool
- **Pool Wallet:** The (sensitive) value is explicitly configured in the pool wallet

user delete

Deletes the key from the configuration.

Example: `delete [--global]`

user add

Add a user and its password and role(s) in the credentials file. If the user already exists, then it is updated.

Non-interactive

Include option `--password-stdin` to read from STDIN. Requires the password file of the user.

Example:

```
ords config user add --password-stdin <USERNAME> roles <ROLES> < userpwd.txt
```

Interactive

Prompts for the password if `--password-stdin` option is omitted.

Example:

```
ords config user add <USERNAME> roles <ROLES>
```

Enter the password:

Confirm password:

user delete

```
ords config user delete <USERNAME>
```

Delete the specified user.

user get

```
ords config user get <USERNAME>
```

Retrieve the specified user.

user list

```
ords config user list
```

Lists the users in the credential file.



See Also:

- [Understanding the Configurable Settings](#)
- [Setting Up the Configuration Folder Structure](#)

3

Configuring Oracle REST Data Services for Multiple Databases

Oracle REST Data Services supports the ability to connect to more than one database. This section describes different strategies for routing requests to the appropriate database.

3.1 About the Request URL

Oracle REST Data Services supports a number of different strategies for routing requests to the appropriate database. All of these strategies rely on examining the request URL and choosing the database based on some kind of match against the URL. It is useful to recap the pertinent portions of a request URL. Consider the following URL:

```
https://www.example.com/ords/sales/f?p=1:1
```

This URL consists of the following sections:

- Protocol: `https`
- Host Name: `www.example.com`
- Context Root: `/ords`

The context root is the location at which Oracle REST Data Services is deployed on the application server.

- Request Path: `/sales/f?p=1.1`

This is the portion of the request URL relative to the context root.

For different applications, it may be important to route requests based on certain prefixes in the request path or certain prefixes in the full request URL.

There are two steps to configuring multiple databases:

1. Configuring the database connection information
2. Configuring which requests are routed to which database

3.2 Configuring Additional Databases

When you first configure Oracle REST Data Services, you configure a default database connection named: `default`. You can create additional database connections using the `install` command.

Tip:

To see full help for the `install` command type:

```
ords install --help
```

For adding additional database connection pool, enter the following command:

```
ords --config <configuration folder> install
```

You get the following prompts:

```
Oracle REST Data Services - Interactive Install
```

```
Enter a number to select the type of installation
  [1] Install or upgrade ORDS in the database only
  [2] Create or update a database pool and install/upgrade ORDS in the
database
  [3] Create or update a database pool only
Choose [1]:
```

Select option 2

```
Enter a number to select the database pool to update, or create an additional
database pool.
```

```
The selected (or created) database pool will be used to install or upgrade
ORDS in the database.
```

```
  [1] default      jdbc:oracle:thin:@//localhost:1521/orcl
  [2] Create an additional database pool
Choose [1]:
```

Select option 2

```
Enter the database pool name:
```

```
Enter db2
```

This becomes the mapping pattern for your new pool. Continue answering the prompts, similar to first time setup of the default pool.



See Also:

[Interactive Command-Line Interface Installation](#)

3.3 Advanced Routing

By default, additional databases are mapped using the database pool name specified while installing ORDS. Alternatively, custom mapping can be performed based on a path prefix or a hostname as described in the following sections.



Note:

If custom mapping is configured, then default mapping with database pool name is not enabled.

**See Also:**[Configuring Additional Databases](#)

3.4 Routing Based on the Request Path

You can create the request routing rules based on the request path using the paths file. To do this, perform the following steps:

1. Open or create a file `<ords_config_folder>/databases/<database_name>/paths`
2. Enter the paths that you want to be served by this database.

Example location of the paths file

```
ords_conf/  
+-- databases/  
|   +-- db1/  
|   |   +-- wallet/  
|   |   +-- paths  
|   |   +-- pool.xml  
|   +-- db2/  
|       +-- wallet/  
|       +-- paths  
|       +-- pool.xml  
+-- globals/
```

Example: `ords_conf/databases/db1/paths`

```
sales
```

Example: `ords_conf/databases/db2/paths`
`ords_conf/databases/db2/paths`

```
support  
/finance/department1
```

The previous rules match all the following requests:

```
http://example.com/ords/sales --> db1  
http://example.com/ords/sales/leads --> db1  
https://www.example.com/ords/sales/forecasting.report?month=jan --> db1 (If  
www.example.com resolves to the same system as example.com.)
```

```
http://example.com/ords/support --> db2  
http://example.com/ords/finance/department1 --> db2
```

3.5 Routing Based on the Request Hostname

You can create request routing rules based on the hostname using the hostnames file. To do this, perform the following steps:

1. Open or create a file `<ords_config_folder>/databases/<database_name>/hostnames`.
2. Add the hostnames you want to be served by this database.

Example location of hostnames files:

```
ords_conf/  
+-- databases/  
|   +-- db1/  
|   |   +-- wallet/  
|   |   +-- hostnames  
|   |   +-- pool.xml  
|   +-- db2/  
|       +-- wallet/  
|       +-- hostnames  
|       +-- pool.xml  
+-- globals/
```

Example: `ords_conf/databases/db1/hostnames`

```
www.example.com  
example.org
```

Example `ords_conf/databases/db2/hostnames`

```
foo.bar.com
```

The preceding rules match the requests to their corresponding database connection as follows:

```
http://www.example.com/ords/ --> db1  
http://example.org:8080/ords/f?p=1:1 --> db1  
https://foo.bar.com/ords/myschema/resource --> db2
```

The preceding rules do not match following requests:

```
http://example.com/ords/ (The hostname is missing the www.)  
http://foo.bar.net/ords/myschema/resource --> db2 (hostname is different)
```

4

Overview of Schema Level Configuration Override

This section describes schema level configuration override feature.

The schema level configuration override feature provides a mechanism to override configuration properties for a REST enabled schema. This allows authorized database users to customize Oracle REST Data Services (ORDS) on the fly. For example, a database user can use this mechanism to enable or disable an optional feature. Only a predefined subset of database schema level configurations can be overridden, global configurations cannot be overridden.

4.1 Features of Schema Level Configuration Override

This section lists the features of schema level configuration override.

The features of schema level configuration override are as follows:

- Database users can override the configuration property for their schema with the ords PL/SQL API.
- Database administrator can override the configuration property for any schema in the database with our ords_admin PL/SQL API.
- The configuration value is cached for 10 mins. You must wait for 10 minutes before the new value becomes effective.

See Also:

- [ORDS PL/SQL Package Reference](#)
- [Oracle REST Data Services Administration PL/SQL Package Reference](#)

4.2 Overidable Configuration Properties

This section lists the overidable configuration properties.

Table 4-1 Overridable Configuration Properties

Key	Description	Scope	Setting Override Value
restEnabledSql.active	Specifies whether the REST-Enabled SQL service is active. REST- Enabled SQL is a prerequisite for SDW or Database Actions. SDW is disabled if the value is set to false.	Schema	<pre>begin ords.set_property(p_key => 'restEnabledSql.active', p_value => 'false'); commit; end; /</pre>
feature.sdw	Specifies whether the Database Actions feature is active.	Schema	<pre>begin ords_admin.set_property(p_ key => 'feature.sdw', p_value => 'false'); commit; end; /</pre>
debug.printDebugToScreen	Specifies whether the error responses include stack trace.	Schema	<pre>begin ords_admin.set_property(p_ key => 'debug.printDebugToScreen' , p_value => 'true'); commit; end; /</pre>

4.3 Use Cases

This section shows some use cases for using the schema level configuration override feature.

Table 4-2 Use Cases of Schema Level Configuration Override feature

Scenario	Solution
An Autonomous Database user wants to debug their ORDS REST Services.	The database user sets the property to <code>true</code> for their schema as shown in the following code snippet: <pre>ords.set_property(p_key => 'debug.printDebugToScreen', p_value => 'true');</pre>
Administrator wants a REST-Enabled schema but not have a REST-Enabled SQL.	The administrator sets the property to <code>true</code> for the relevant schema as shown in the following code snippet: <pre>ords_admin.set_property(p_schema => 'REST_PRODUCTION', p_key => 'restEnabledSql.active', p_value => 'true');</pre>
Administrator wants to completely disable SDW for a database.	The administrator sets the property for every schema as shown in the following code snippet: <pre>ords_admin.set_property(p_schema => 's1', p_key => 'feature.sdw', p_value => 'false'); ords_admin.set_property(p_schema => 's2', p_key => 'feature.sdw', p_value => 'false');</pre>
REST-Enabled schema wants to clear a property value that was previously set.	The database user runs the following command: <pre>ords.unset_property(p_key => 'debug.printDebugToScreen');</pre>

4.4 Scope and Visibility Matrix

This section describes how to configure the visibility and the scope for the database and schema users.

Table 4-3 Scope and Visibility Matrix

User Type	Database	Schema
Public	<ul style="list-style-type: none"> • Only administrator can set a value that applies to all the schemas • USER_ORDS_PROPERTIES view shows the value, if it is set • Unprotected REST endpoint shows the value if it is set • Currently used for login message and SDW header and footer 	<ul style="list-style-type: none"> • Administrator and non-administrator user can set a value per schema • There is no default value that automatically applies to a schema • USER_ORDS_PROPERTIES view shows the value if it is set • Unprotected REST endpoint shows the value if it is set
PROTECTED	<ul style="list-style-type: none"> • Only the administrator can set a value that applies for all the schemas • USER_ORDS_PROPERTIES view shows the value if it is set • Unprotected REST endpoint does not show value if it is set 	<ul style="list-style-type: none"> • Administrator and non-administrator user can set a value per schema • There is no default value that automatically applies to a schema • USER_ORDS_PROPERTIES view shows the value if it is set • Unprotected REST endpoint does not show the value if it is set
PRIVATE	<ul style="list-style-type: none"> • Only administrator can set the value that applies for all the schemas • USER_ORDS_PROPERTIES view does not show a value if it is set • Unprotected REST endpoint does not show a value if it is set 	<ul style="list-style-type: none"> • Only the administrator can set a value per schema • There is no default value that automatically applies to a schema • USER_ORDS_PROPERTIES view does not show the value if it is set • Unprotected REST endpoint does not show the value if it is set

5

Miscellaneous Configuration Options of Oracle REST Data Services

This section explains how to configure Oracle REST Data Services for connecting to multiple databases for routing requests, and it refers to other documentation sources for other configuration information.

Note:

You must restart Oracle REST Data Services after making any configuration changes. For ensuring High Availability, Oracle recommends that you use a load balancer in front of multiple ORDS instances, so that rolling restarts can be achieved.

Topics:

- [Using the REST Enabled SQL Service with MySQL Database](#)
- [Support for Oracle RAC Fast Connection Failover](#)
- [Configuring REST-Enabled SQL Service Settings](#)
- [Configuring the Maximum Number of Rows Returned from a Query](#)
- [Configuring ICAP Server Integration for Virus Scan](#)
- [Configuring ORDS with Kerberos Setup](#)
- [Configuring the Custom Error Pages](#)
- [Configuring ORDS Metadata Cache](#)

5.1 Using the REST Enabled SQL Service with MySQL Database

This section describes the ORDS feature which is only supported with MySQL database running on Oracle Cloud Infrastructure.

You can setup your own ORDS instance to use the REST Enabled SQL Service with a MySQL database over JDBC. The connection details are specified similar to how you specify for other ORDS connection pool. For MySQL JDBC connections, the `db.connectionType` is always `customurl`. The `db.customURL` property must be a valid JDBC connection string for the database. The host machine running the database must be accessible to the host machine with ORDS instance running. The MySQL database accounts to be used must be configured to permit logins from the host machine with ORDS instance running. ORDS supports connecting to Oracle hosted MySQL database servers such as the MySQL Data Service or an Oracle Compute Instance running MySQL server.

5.1.1 Understanding the Database Credentials Source Setting

For each request received, ORDS creates a JDBC connection for executing the SQL statements in the request. To create the JDBC connection, ORDS can be configured to use the pool connection details, or uses the basic authorization credentials in the request. The

credentials are specified using the `db.credentials` source configuration property. Possible values are `POOL` (default value) or `REQUEST`.

 **Note:**

To access the REST Enabled SQL Service, a client must have the ORDS SQL developer role.

If you are using the default value as `POOL`, then the credentials in the pool configuration are used for processing any SQL statements in the request. However, the client must provide credentials for an identity management system to authorize them and assign them the SQL developer role. Only then the client can access to the REST Enabled SQL service.

If the value is set to `REQUEST`, then the username and password specified in the pool configuration are still required. However, these credentials are only used to verify the connection details in the pool, when the pool is used for the first time. The username and password in the basic authorization header are used to make a new JDBC connection with the target database. If a connection is established, then the client is considered to have the SQL developer role. This authorizes them to invoke the REST Enabled SQL service. The new JDBC connection is used during the request lifecycle and then closed.

5.1.2 Configuring a Pool for MySQL Database

Pool configuration is required to use ORDS with MySQL database. You can configure the pool through the ORDS command line interface.

You must configure ORDS to be able to use a MySQL database with Oracle REST Data Services (ORDS) running in a customer managed environment. Depending on where you install Oracle REST Data Services for your customer managed environment, do one of the following:

- If your customer managed environment for Oracle REST Data Services runs in Oracle Cloud Infrastructure, then use an Oracle YUM repository and perform a YUM install of ORDS.
- If your customer managed environment for Oracle REST Data Services runs in some other environment, then download ORDS from the [Oracle REST Data Services Download](#) page.

To use ORDS with MySQL database, you do not need any installation to the database, you only need to do a pool configuration. You can configure the pool through the ORDS command line interface.

To configure the pool for MySQL database, perform the following steps:

 **Note:**

The credentials in a request are used to execute the SQL statements. The `db.username` specified in the MySQL database is the user who has all the permissions to create a connection and is used to verify the overall pool configuration details.

```
ords config --db-pool mysql set db.connectionType customurl
ords config --db-pool mysql set db.customURL "jdbc:mysql://10.0.1.23/?
```

```
sslMode=REQUIRED"
ords config --db-pool mysql set db.username
user_only_has_permission_to_connect_and_nothing_more
ords config --db-pool mysql set db.credentialsSource request
ords config --db-pool mysql set restEnabledSql.active true
ords config --db-pool mysql secret db.password
```

In the preceding example,

- Any relevant properties for the JDBC driver can be specified in the `db.customURL` property. In the preceding example, `db.customURL` value, `sslMode` is set to `REQUIRED`, rather than the default value `PREFERRED`, to ensure a secure connection between ORDS and the MySQL server.
- The database pool is called `mysql`. However, the pool can have any name. The default pool can be configured as a MySQL connection pool. You can have multiple pools defined for as many MySQL databases you wish to use.
- The `db.username` specified is a MySQL database user who has sufficient permissions to create a connection. This database account is used to verify the overall pool configuration details.

5.1.2.1 Configuring ORDS for Supported Containers

This section describes how to use ORDS with connection pool configuration in a supported container for a MySQL database.

Specifying the Configuration Location

When you are running ORDS in a standalone mode using the `ords serve` command, there are options for specifying the configuration directory location. When deploying the `ords.war` in a supported container such as Apache Tomcat or WebLogic Server, the configuration directory location must be specified by setting the `config.url` system property. The mechanism for doing this varies for different container products.

- To set the `config.url` system property before starting Apache Tomcat, run the following command:

```
export JAVA_OPTS="-Dconfig.url=/scratch/my_ords_config"
```

- To set `config.url` system property before starting WebLogic Server, run the following command:

```
export JAVA_OPTIONS="-Dconfig.url=/scratch/my_ords_config"
```

- Alternatively, use the `ords war` command to create a deployable web application archive file which has the `config.url` context parameter explicitly set and any `jar` files from `lib/ext` folder are included.

MySQL JDBC Jar for ORDS

ORDS does not distribute a MySQL JDBC jar. For ORDS to create a JDBC connection to a MySQL database, the relevant JDBC jar must be in the runtime classpath. Use `OCI YUM mysql-connector-java` or download MySQL Connector/J from <https://www.mysql.com/> and then copy the `jar` file to the relevant location for one of the server modes: Standalone, Apache Tomcat, or WebLogic Server.

 **Note:**

The minimum required version of MySQL Connector/J is 8.0.27.

The OCI YUM RPM distribution of ORDS creates a symbolic link to the OCI YUM `mysql-connector-java` JDBC jar.

OCI YUM RPM

```
-- Install MySQL Connector/J community edition
sudo yum install mysql-connector-java

-- Confirm JDBC jar is installed
ls -l /usr/share/java/mysql-connector-java.jar

-- Install ORDS from OCI YUM repository
sudo yum install ords

-- Note that ORDS RPM install will create a symbolic link to ORDS
installation lib/ext/ directory
ls -l /opt/oracle/ords/lib/ext/
```

5.1.2.1.1 ORDS Running in Standalone Mode

To be in the runtime classpath when running ORDS in a standalone mode, the MySQL JDBC jar must be first added to the Extension folder. The Extension folder is `lib/ext` directory in the ORDS distribution and is created through the OCI YUM RPM installation process that is outlined in the preceding section.

5.1.2.1.2 ORDS Deployed in Apache Tomcat

 **Note:**

When you are using Apache Tomcat, to avoid getting `java.sql.SQLException: No suitable driver` errors, it is necessary to explicitly set the JDBC Driver classname in the pool.

To set the JDBC driver classname in the pool, run the following command:

```
ords config --db-pool mysql set jdbc.driverName com.mysql.cj.jdbc.Driver
```

To be in the runtime classpath when ORDS is deployed in Apache Tomcat, the MySQL JDBC jar must be added to the server classpath or in the deployed web application. Adding the jar to the server classpath can be achieved in a number of ways, the most common method is adding the jar file to the `$CATALINA_HOME/lib` directory.

See Apache Tomcat documentation for more options and guidelines to determine the most appropriate deployment environment.

To include the JDBC jar in the deployed web application, ensure that it is in the `lib/ext/` folder and uses the `ords war` command to create a deployable web application archive file

which has the `config.url` context parameter explicitly set and any `jar` files from `lib/ext` folder included.

 **See Also:**

[Apache Tomcat 8](#)

5.1.2.1.3 ORDS Deployed in Weblogic Server

To be in the runtime classpath when ORDS is deployed in WebLogic Server, the MySQL JDBC `jarfile` must be added to the server classpath or in the deployed web application. One approach to add the `jar` file to the server classpath is to specify the `jar` location in the `WEBLOGIC_CLASSPATH` environment variable in the `commEnv.cmd/sh` script.

See WebLogic Server documentation for more options and guidelines to determine the most appropriate deployment environment.

To include the JDBC `jar` in the deployed web application, ensure that it is in the `lib/ext/` folder and uses the `ords war` command to create a deployable web application archive file which has the `config.url` context parameter explicitly set and any `jar` files from `lib/ext` included.

 **See Also:**

[Understanding WebLogic Server Deployment](#)

5.2 Configuring Jetty in ORDS Standalone Mode

This section describes how to configure the Eclipse Jetty Server used in ORDS standalone mode.

In a standalone mode, ORDS uses an embedded Eclipse Jetty server. The default configuration of Jetty is optimised for the most common ORDS use cases. This Jetty server configuration can be modified using certain ORDS settings as defined in Java system properties and also in the Jetty XML configuration in the following directory: `{configuration.directory}/global/standalone/etc.`

 **Note:**

These settings have no effect when ORDS is deployed in a container such as Apache Tomcat or Oracle WebLogic Server.

 **See Also:**

- [Eclipse Jetty server](#)
- [Understanding the Configurable Settings](#)

5.2.1 Using Java System Properties

This section describes how to provide a Java system property to modify the settings in the Jetty configuration used by ORDS.

To specify a system property, you must use the `JDK_JAVA_OPTIONS` environment variables before starting ORDS in a standalone mode.

Example:

Example 5-1 Using the `JDK_JAVA_OPTIONS`

```
export JDK_JAVA_OPTIONS='-Dthreads.max=300 -Djetty.request.header.size=8192'
ords --config /path/to/config/ serve
```



See Also:

JDK Java Options

5.2.2 Java System Properties

This section lists the Java system properties.

Table 5-1 Java System Properties

Java System Property Name	Description	Default Value
<code>threads.min</code>	Specifies the minimum number of threads in the Jetty Server (id="Server") thread pool.	10
<code>threads.max</code>	Specifies the maximum number of threads in the Jetty Server (id="Server") thread pool.	200
<code>threads.timeout</code>	Specifies the maximum thread idle time in milliseconds for the Jetty Server (id="Server") thread pool. Threads that are idle for longer than this period may be stopped.	60000
<code>jetty.send.xpoweredBy</code>	Specifies the <code>SendXPoweredBy</code> setting in the Jetty <code>HttpConfiguration</code> (id="httpConfig")	false
<code>jetty.output.buffer.size</code>	Specifies the <code>OutputBufferSize</code> setting in the Jetty <code>HttpConfiguration</code> (id="httpConfig")	32768
<code>jetty.request.header.size</code>	Specifies the <code>RequestHeaderSize</code> setting in the Jetty <code>HttpConfiguration</code> (id="httpConfig")	65536

Table 5-1 (Cont.) Java System Properties

Java System Property Name	Description	Default Value
jetty.response.header.size	Set the ResponseHeaderSize setting in the Jetty HttpConfiguration (id="httpConfig")	8192
jetty.send.server.version	specifies the SendServerVersion setting in the Jetty HttpConfiguration (id="httpConfig")	false
jetty.send.date.header	Specifies the SendDateHeader setting in the Jetty HttpConfiguration (id="httpConfig")	false
jetty.dump.start	Specifies the DumpAfterStart setting in the Jetty Server (id="Server")	false
jetty.dump.stop	Specifies the DumpAfterStop setting in the Jetty Server (id="Server")	false

5.2.3 Using Jetty XML Configuration Files

This section describes how to configure Jetty server for additional functionality using the Jetty XML configuration files.

When ORDS is running in a standalone mode, the Eclipse Jetty Home is `${configuration.directory}/global/standalone/`. The Jetty XML syntax can be used to configure the Jetty Server for additional functionality by placing configuration XML files in the Jetty Home `etc` directory. The capability to do this is provided through the Eclipse Jetty server product.



See Also:

Eclipse Jetty documentation

Examples

This section provides examples of augmenting Eclipse Jetty Server behaviour for additional functionality. The following examples show how you can change Eclipse Jetty Server behaviour and not the ORDS behaviour. The location for Jetty XML files to be used by ORDS at the startup in standalone mode is:

```
${configuration.directory}/global/standalone/etc/
```

Example 5-2 Using a specific access log format

When the configuration setting `standalone.access.log` is provided, ORDS can produce an access log.

```
/global/standalone/etc/jetty-access-log.xml
<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://www.eclipse.org/
jetty/configure.dtd">
<Configure id="Server" class="org.eclipse.jetty.server.Server">
  <Ref id="Handlers">
    <Call name="addHandler">
      <Arg>
        <New id="RequestLog"
class="org.eclipse.jetty.server.handler.RequestLogHandler">
          <Set name="requestLog">
            <New id="RequestLogImpl"
class="org.eclipse.jetty.server.CustomRequestLog">
              <Arg>/ords/ords-access.log</Arg>
              <Arg>{%remote}a - %u %t "%r" %s %O "%{Referer}i" "%{User-
Agent}i"</Arg>
            </New>
          </Set>
        </New>
      </Arg>
    </Call>
  </Ref>
</Configure>
```

Example 5-3 Always returning a certain header in the response

Although this can also be achieved through a Load Balancer or Reverse Proxy in front of ORDS. If you want a specific header to be returned in every response from the ORDS server. Then use the following sample code snippet:

```
/global/standalone/etc/jetty-response.xml
<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://www.eclipse.org/
jetty/configure.dtd">
<Configure id="Server" class="org.eclipse.jetty.server.Server">
  <Call name="insertHandler">
    <Arg>
      <New class="org.eclipse.jetty.rewrite.handler.RewriteHandler">
        <Get id="Rewrite" name="ruleContainer" />
        <Call name="addRule">
          <Arg>
            <New id="header"
class="org.eclipse.jetty.rewrite.handler.HeaderPatternRule">
              <Set name="pattern">*</Set>
              <Set name="name">Strict-Transport-Security</Set>
              <Set name="value">max-age=31536000;includeSubDomains</Set>
            </New>
          </Arg>
        </Call>
      </New>
    </Arg>
  </Call>
</Configure>
```

```
</Call>
</Configure>
```

5.3 Support for Oracle RAC Fast Connection Failover

Oracle REST Data Services support the Fast Connection Failover (FCF) feature of Oracle Real Application Clusters (Oracle RAC).

Oracle REST Data Services runs with the Universal Connection Pool (UCP) in all the Application Server environments that it supports, such as WebLogic, Tomcat. UCP in turn supports Fast Connection Failover. To enable FCF, Oracle Notification Service (ONS) must be enabled. To enable ONS, add entries to the list of properties in the Oracle REST Data Services `settings.xml` configuration file as shown in the following code snippet:

```
<entry key="jdbc.enableONS">true</entry>
<entry key="jdbc.ONSConfig">nodes=racnode1:4200,racnode2:4200\nwalletfile=/
oracle11/onswalletfile</entry>
```

ONS is the messaging facility used to send the Fast Application Notification (FAN) events. When ONS is enabled, Oracle REST Data Services automatically enables FCF. To Enable specific FCF capabilities such as fail over or other advanced FCF capabilities such as load balancing, you need to add entries in the configuration file for the custom connection as shown in the following code snippet:

```
<entry key="db.connectionType">customurl</entry>
<entry key="db.customURL">jdbc:oracle:thin:@(DESCRIPTION=(FAILOVER=ON)
  (ADDRESS_LIST=
    (LOAD_BALANCE=ON) (ADDRESS=(PROTOCOL=TCP) (HOST=prod_scan.example.com)
  (PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=ISPRD)))</entry>
```

After updating the `defaults.xml` configuration file, Oracle REST Data Services need to be restarted for the changes to take effect.

UCP supports Fast Connection Failover. FCF listens and responds to FAN events to deal with the following two scenarios:

- **Unplanned outages:** When RAC detects an instance failure, it generates a FAN Down event which FCF picks up. FCF then terminates all connections to the failed instance and directs all future requests to the surviving RAC instances.
- **Planned outages:** For instance, when a Database Administrator (DBA) wants to gracefully shut down a RAC instance for performing some maintenance activity. The instance shutdown generates a FAN Planned Down event which FCF picks up. FCF then directs all new requests to other RAC instances and **drains** or allows currently active transactions to complete.

Note:

Long running transactions may need to be terminated forcefully.

5.4 Configuring ORDS with Kerberos Setup

This section explains how ORDS can be configured to reference a Kerberos file-based ticket cache and make a connection to an Oracle Database Kerberos authenticated user with the ORDS Runtime Privilege.

To configure ORDS with Kerberos setup, perform the following steps:

1. Create a new user using external authentication
2. Set up the environment variables
3. Provide a valid ticket
4. Add ORDS pool settings

1. Create a New User using External Authentication

Create a new Oracle Database user using external authentication (Kerberos) and provision the user as an ORDS runtime user.

```
CREATE USER ORDS_PUBLIC_KRBUSER IDENTIFIED EXTERNALLY AS
'<kerberos_principal_name>';
GRANT CONNECT TO "ORDS_PUBLIC_KRBUSER";
BEGIN
    ORDS_ADMIN.PROVISION_RUNTIME_ROLE (
        p_user => 'ORDS_PUBLIC_KRBUSER',
        p_proxy_enabled_schemas => TRUE);
END;
/
```

2. Set up the Environment Variables

 **Note:**

Ensure that you have a Kerberos configuration file `krb5.conf` and a file-based ticket cache

Set up the following environment variables:

```
export KRB5_CONFIG=<path to krb5.conf>
export KRB5CCNAME=<path to credential cache>
```

3. Provide a Valid Ticket

Provide a valid ticket in the ticket cache to get authenticated when connecting to the Oracle Database.

```
kinit <principal>
```

4. Add ORDS Pool Settings

Add the following pool settings to the `pool.xml` file using the ticket in the ticket cache:

```
<entry key="oracle.net.authentication_services">(KERBEROS5)</entry>
<entry key="oracle.net.kerberos5_mutual_authentication">>true</entry>
```

For example, to run ORDS in standalone mode with Kerberos, execute the following command:

```
export JDK_JAVA_OPTIONS=-Djava.security.krb5.conf=$KRB5_CONFIG
ords serve
```

5.5 Authorizing Oracle REST Data Services to Access Oracle Data Guard Protected Users

To access the database schema objects that are protected by an Oracle Data Vault Realm, it is necessary to grant a proxy user authorization to the Oracle REST Data Services Public User.

The following example authorizes the Oracle REST Data Services Public User, `ORDS_PUBLIC_USER` to proxy the database `HR` user:

```
begin
  DBMS_MACADM.AUTHORIZE_PROXY_USER('ORDS_PUBLIC_USER', 'HR');
end;
/
```

5.6 Configuring REST-Enabled SQL Service Settings

This section explains how to configure the REST- Enabled SQL service.

Note:

Enabling the REST- Enabled SQL service enables authentication against the Oracle REST Data Service enabled database schemas. This makes the database schemas accessible over HTTPS, using the database password. Oracle highly recommends that you provide strong secure database passwords

REST- Enabled SQL service is a feature of Oracle REST Data Service. By default, the REST Enabled SQL service is turned off. To enable the REST- Enabled SQL service and the REST- Enabled SQL Export service, perform the following steps:

1. Run the following command:

```
ords --config <configuration_folder> config [--db-pool <pool_name>] set restEnabledSql.active true
```
2. Restart Oracle REST Data Services.

5.7 Configuring the Maximum Number of Rows Returned from a Query

To configure maximum number of rows returned from a query, perform the following steps:

1. Run the following command:

```
ords --config <configuration_folder> config set [--db-pool <pool_name>]  
misc.pagination.maxRows <number>
```

 **Note:**

The default value for `misc.pagination.maxRows` is 10000.

2. Restart Oracle REST Data Services.

5.8 Configuring ICAP Server Integration for Virus Scan

This section explains how to configure ORDS to integrate with ICAP server for virus scan.

ORDS PL/SQL gateway supports the offloading of virus scanning responsibilities to an Internet Content Adaptation Protocol (ICAP) compliant virus scan server when the files are uploaded. The hostname and port of the virus scan server is specified in the `icap.server`, `icap.port`, and `icap.secure.port` global configuration properties.

APEX uses ORDS PL/SQL gateway. Once configured, this ICAP integration is also applied to file uploads in APEX.

To configure ORDS to integrate with ICAP server, perform the following steps:

1. Run the following command:

```
ords --config <configuration_folder> config [--db-pool <pool_name>] set  
icap.port <number> ords --config <configuration_folder> config [--db-pool  
<pool_name>] set icap.server <name_or_ip>
```
2. Restart Oracle REST Data Services.

ICAP server must support the following requirements:

- ICAP version 1.0
- Antivirus service named AVSCAN
- Antivirus service that supports action=SCAN
- Previews of at least 4 bytes
- Return header named X-Infection

Once configured, when a file is uploaded through PL/SQL Gateway, ORDS makes a request similar to the following:

```
RESPMOD icap://<icap_server>:<icap_port>/AVSCAN?action=SCAN ICAP/1.0  
Host: <icap_server>:<icap_port>  
Preview: 4  
Allow: 204  
Encapsulated: req-hdr=0 res-hdr=153 res-body=200
```

5.9 Configuring the Custom Error Pages

This section explains how to configure a custom error page instead of the error page generated by Oracle REST Data Services.

To configure a custom error page,

1. Run the following command:

```
ords --config /path/to/conf config set error.externalPath /path/to/error/
pages/folder/
```

Where:

`/path/to/error/pages/folder` is the path to a folder containing files that define the error pages. The files are stored in `{status}.html` format. Where, `{status}` is the HTTP status code for which you want to create a custom error page.

2. Restart Oracle REST Data Services

Example 5-4 Configuring custom error page for “HTTP 404” status code

To configure a custom error page for the “HTTP 404 – Not Found” status, perform the following steps:

1. Create a file named `404.html`.
2. Save it under `/usr/local/share/ords/error-pages/` folder.
3. Configure the `error.externalPath` parameter to point to `/usr/local/share/ords/error-pages/` folder.
4. Restart Oracle REST Data Services.

5.10 Managing ORDS Administrator Privilege

Access to the `ORDS_ADMIN` PL/SQL package is provisioned through the `ORDS_ADMINISTRATOR_ROLE`. This role can be provisioned through the `ORDS_ADMIN` package to create additional ORDS administrators.

5.10.1 Provisioning `ORDS_ADMINISTRATOR_ROLE` to a User

This section describes how to provision `ORDS_ADMINISTRATOR_ROLE` role to a user.

You can provision `ORDS_ADMINISTRATOR_ROLE` role to a user by using either the database `GRANT` command or through the `ORDS_ADMIN.PROVISION_ADMIN_ROLE` PL/SQL method (as an ORDS Administrator).

Example 5-5 Using Grant command

```
GRANT ORDS_ADMINISTRATOR_ROLE TO HR_ADMIN;
```

Example 5-6 Using `ORDS_ADMIN` package method

```
BEGIN
  ORDS_ADMIN.PROVISION_ADMIN_ROLE (
    p_user => 'HR_ADMIN');
END;
/
```

5.10.2 Unprovisioning `ORDS_ADMINISTRATOR_ROLE` from a User

This section describes how to unprovision `ORDS_ADMINISTRATOR_ROLE` from a user.

As an ORDS administrator, you can unprovision `ORDS_ADMINISTRATOR_ROLE` from a user by either using the database `REVOKE` command or through the `ORDS_ADMIN.UNPROVISION_ROLES` PL/SQL method.

Example 5-7 Using REVOKE command

```
REVOKE ORDS_ADMINISTRATOR_ROLE FROM HR_ADMIN;
```

Example 5-8 Using ORDS_ADMIN package method

```
BEGIN
  ORDS_ADMIN.UNPROVISION_ROLES (
    p_user => 'HR_ADMIN',
    p_administrator_role => TRUE);
END;
/
```

5.11 Managing ORDS Runtime Privilege

The `ORDS_RUNTIME_ROLE` database role allows a user to act as a runtime user. A runtime user can manage and configure the runtime connection resources required by an ORDS service instance. The `ORDS_PUBLIC_USER` is one such database user. When additional runtime users are provisioned, it is possible to configure discrete ORDS service instances with different destination addresses and connection pools but hosted on the same Oracle database container.

It is recommended not to re-use a runtime user for any other purpose as it accumulates the grants necessary to proxy to other users. A runtime user only requires the `CREATE SESSION` privilege in addition to the `ORDS_RUNTIME_ROLE` role.

5.11.1 Provisioning ORDS_RUNTIME_ROLE to a User

This section describes how to provision `ORDS_RUNTIME_ROLE` role to a user.

As an ORDS administrator, you can provision `ORDS_RUNTIME_ROLE` role to a user by using either the database `GRANT` command or through the `ORDS_ADMIN.PROVISION_ADMIN_ROLE` PL/SQL method.

Example 5-9 Using Grant command

```
GRANT ORDS_RUNTIME_ROLE TO ORDS_PUBLIC_USER_2;
```

Example 5-10 Using ORDS_ADMIN package method

```
BEGIN
  ORDS_ADMIN.PROVISION_RUNTIME_ROLE (
    p_user => 'ORDS_PUBLIC_USER_2');
END;
/
```

5.11.2 Unprovisioning ORDS_RUNTIME_ROLE from a User

This section describes how to unprovision the `ORDS_RUNTIME_ROLE` role from a user

As an administrator, you can unprovision the `ORDS_RUNTIME_ROLE` from a user, by either using the database `REVOKE` command or through the `ORDS_ADMIN.UNPROVISION_ROLES` PL/SQL method.

Example 5-11 Using REVOKE command

```
REVOKE ORDS_RUNTIME_ROLE FROM ORDS_RUNTIME_USER_2;
```

Example 5-12 Using ORDS_ADMIN package method

```
BEGIN
  ORDS_ADMIN.UNPROVISION_ROLES (
    p_user => 'ORDS_RUNTIME_USER_2',
    p_runtime_role => TRUE);
END;
/
```

5.12 Using OAuth2 in Non-HTTPS Environments

RESTful Services can be protected with the OAuth2 protocol to control access to nonpublic data. To prevent data snooping, OAuth2 requires all requests involved in the OAuth2 authentication process to be transported using HTTPS. The default behavior of Oracle REST Data Services is to verify that all OAuth2 related requests have been received using HTTPS. It will refuse to service any such requests received over HTTP, returning an HTTP status code of 403 Forbidden.

This default behavior can be disabled in environments where HTTPS is not available as follows:

1. Locate the folder where the Oracle REST Data Services configuration is stored, for example: `/path/to/conf`
2. Execute the following command:
`ords --config /path/to/conf config set security.verifySSL false`
3. Restart Oracle REST Data Services if it is running.

Note that it is only appropriate to use this setting in development or test environments. It is never appropriate to use this setting in production environments because it will result in user credentials being passed in clear text.

Note:

Oracle REST Data Services must be restarted after making configuration changes. See your application server documentation for information on how to restart applications.

5.13 Configuring ORDS Metadata Cache

This section explains how to configure the ORDS Metadata Cache.

As the number of REST services grow, the overhead of querying the database for corresponding metadata can have a negative impact on the overall service performance and throughput. Overtime, the queries for `ORDS_METADATA` views take longer time to complete. These queries are executed for every request. The ORDS metadata cache can help improve the overall response time for REST services when the number of services grow to an extent that querying the `ORDS_METADATA` views for every request becomes expensive. The ORDS metadata cache can temporarily hold a copy of privilege and module metadata in memory to reduce the number of database queries performed when a REST service request is received. The cache is disabled by default so that the changes made to the metadata are applied immediately for any subsequent request.

Table 5-2 Configuration Properties for ORDS Metadata Cache

Property	Data Type	Default Value	Description
<code>cache.metadata.enabled</code>	Boolean	<code>false</code>	Specifies a setting to enable or disable metadata caching.
<code>cache.metadata.timeout</code>	Duration	30s	Specifies a setting that determines for how long the metadata record remains in the cache. Longer the duration, it takes longer to view the applied changes.

6

Deploying and Monitoring Oracle REST Data Services

This chapter discusses how to deploy and monitor ORDS.

6.1 Deploying Oracle REST Data Services

To deploy Oracle REST Data Service, you can choose one of the following options:

- [Serve Commands for Running in Standalone Mode](#)
- [Oracle WebLogic Server](#)
- [Apache Tomcat](#)
- [Oracle Cloud Infrastructure](#)

6.1.1 Serve Commands for Running in Standalone Mode

This section describes the serve command options for running in standalone mode.

Although Oracle REST Data Services supports the Java EE application servers, you also have the option of running it in standalone mode. This section describes how to run Oracle REST Data Services in a standalone mode.

Standalone mode is suitable for development use and is supported in production deployments. Standalone mode, however, has minimal management capabilities when compared to most Java EE application servers and may not have adequate management capabilities for production use in some environments.

6.1.1.1 Non-Interactive Serve CLI

The non-interactive serve CLI reads the standalone settings from the global `settings.xml` file located in the user-specified configuration folder or the default configuration folder. Some standalone settings contain default values if it is a required setting (for example: http port, context path, and so on.) that does not exist.

```
$ ords --config <CONFIG_PATH> serve
$ ords --config <CONFIG_PATH> serve --port 8777 --apex-images /path/to/apex/
images
```

Example:

Create the Standalone Settings



Note:

You can configure and start ORDS using the interactive install command.

- You can create, update, or delete the standalone settings using the following config CLI command:

```
ords --config <your configuration folder> config set standalone.context.path /
ords
```

- If you are using APEX, then you must provide the path to APEX images as follows:

```
ords --config <your configuration folder> config set standalone.static.path /
path/to/apex/images
```

- If you are using HTTP protocol, then specify the following command:

```
ords --config /path/to/test/config config set standalone.http.port 8443
```

- If you are using a certificate, then specify the following command:

```
ords --config <your configuration folder> config set standalone.https.port
8443
ords --config <your configuration folder> config set
standalone.https.cert /path/to/certificate/mycert.crt
ords --config <your configuration folder> config set
standalone.https.cert.key /path/to/certificate/mycert.key
```

6.1.1.2 Serve Command Options

Long Option	ShortOption	Description
--apex-images-context-path <STATIC_CONTEXT_PATH>		Specifies the context path where APEX static resources are located, defaults to /i.
--certificate <SSL_CERTIFICATE_PATH>		Sets the certificate file for HTTPS, implies --secure/--ssl SSL certificate path. If you are providing the SSL certificate, you must specify the certificate location.
--config <CONFIG_PATH>	c <CONFIG_PATH>	Specifies the path to the folder containing the configuration files.
--context-path <CONTEXT_PATH>		Sets the first part of the path of the URL from which ORDS is served. Defaults to /ords
--document-root <DOCUMENT_ROOT_PATH>		Specifies the path from which static files are served at the root of the server. The context path takes priority over static files that have the same name.
--ip-addresses <INTERFACE_IP>		Specifies ORDS which interface to bind to, use 0.0.0.0 (The default) to bind to all the network interfaces.

Long Option	ShortOption	Description
<code>--key <SSL_KEY_PATH></code>		Sets the certificate key to use for HTTPS, implies <code>--secure/--ssl</code> , requires <code>--certificate</code> . Specifies the SSL certificate key path. If you are providing the SSL certificate, you must specify the certificate key location.
<code>--port <PORT></code>		Specifies the HTTP listen port, default value is 8080. Set the port on which the server should run.
<code>--secure</code>		Specifies HTTPS listen port, default value is 8443. Must specify options <code>--secure --port <PORT></code> to use HTTPS. Starts the server with HTTPS capabilities

6.1.1.3 Converting a Private Key to DER (Linux and Unix)

Usually, you would have created a private key and a Certificate Signing Request before obtaining your signed certificate. The private key needs to be converted into DER in order for Oracle REST Data Services to read it properly.

For example, assume that the original private key was created using the OpenSSL tool with a command similar to either of the following:

```
openssl req -new -newkey rsa:2048 -nodes -keyout yourdomain.key -out
yourdomain.csr
```

or

```
openssl genrsa -out private.em 2048
```

In this case, you must run a command similar to the following to convert it and remove the encryption: `openssl pkcs8 -topk8 -inform PEM -outform DER -in yourdomain.key -out yourdomain.der -nocrypt`

```
openssl pkcs8 -topk8 -inform PEM -outform DER -in yourdomain.key -out
yourdomain.der -nocrypt
```

After doing this, you can include the path to `yourdomain.der` when you are prompted using the `install` interactive command, or you can modify the following entries in `conf/ords/standalone/standalone.properties`:

```
ssl.cert=<path to yourdomain.crt>
ssl.cert.key=<path to yourdomain.der>
ssl.host=yourdomain
```

Also, ensure that `jetty.secure.port` is set.

6.1.1.4 Configuring a Doc Root for Non-APEX Static Resources

You can configure a doc root for standalone mode to deploy static resources that are outside the `/i` folder that is reserved for APEX static resources.

To do so, execute the following command:

```
ords --config /path/to/conf config set standalone.doc.root /var/www/html.
```

The preceding example makes any resource located within `/var/www/html` available under `http://server:port/`. For example, if the file `/var/www/html/hello.txt` exists, it will be accessible at `http://server:port/hello.txt`.

The value specified for `standalone.doc.root` is stored in `${config.dir}/global/settings.xml` in the `standalone.doc.root` property. If a custom doc root is not specified, then the default doc root value of `${config.dir}/global/doc_root` is used. Any file placed within this folder is available at the root context.

This approach has the following features and considerations:

- HTML resources can be addressed without their file extension. For example, if a file named `hello.html` exists in the doc root, it can be accessed at the URI `http://server:port/hello`.
- Attempts to address a HTML resource with its file extension are redirected to the location without an extension. For example, if the URI `http://server:port/hello.html` is accessed, then the client is redirected to `http://server:port/hello`.

The usual practice is to serve HTML resources without their file extensions, so this feature facilitates that practice, while the redirect handles the case where the resource is addressed with its file extension.

- Index pages for folders are supported. If a folder contains a file named `index.html` or `index.htm`, then that file is used as the index page for the folder. For example, if `/var/www/html` contains `/abc/xyz/index.html`, then accessing `http://server:port/abc/xyz/` displays the contents of `index.html`.
- Addressing a folder without a trailing slash causes a redirect to the URI with a trailing slash. For example, if a client accesses `http://server:port/abc/xyz`, then the server issues a redirect to `http://server:port/abc/xyz/`.
- Resources are generated with weak etags based on the modification stamp of the file and with a Cache Control header that causes the resources to be cached for 1 hour.

6.1.2 Oracle WebLogic Server

This section describes how to deploy Oracle REST Data Services on Oracle WebLogic Server. It assumes that you have completed the installation process and are familiar with Oracle WebLogic Server. If you are unfamiliar with domains, managed servers, deployment, security, users and roles, refer to your Oracle WebLogic Server documentation.

Topics:

- [About Oracle WebLogic Server](#)
- [Configuring Oracle APEX Images](#)
- [Launching the Administration Server Console](#)
- [Deploying ORDS on Oracle WebLogic Server](#)
- [Configuring WebLogic to Handle HTTP Basic Challenges Correctly](#)
- [Verifying the State and Health of ords and i](#)

6.1.2.1 About Oracle WebLogic Server

You can download Oracle WebLogic Server from Oracle Technology Network.

To learn more about installing Oracle WebLogic Server, see *Oracle Fusion Middleware Getting Started With Installation for Oracle WebLogic Server* and *Oracle Fusion Middleware Installation Guide for Oracle WebLogic Server*.



See Also:

Oracle Fusion Middleware Software Downloads

6.1.2.2 Configuring Oracle APEX Images

This section is only applicable if you are using Oracle APEX. If you are not using Oracle APEX, then you can skip this section that provides details about configuring Oracle APEX Images.

If you are using Oracle APEX, then the Oracle APEX image files for that Oracle APEX version must also be accessible to the end user. It is recommended that you configure the Oracle APEX instance wherever applicable to use the resources from the Oracle Content Network. The use of the Oracle CDN may not be feasible when you are using Oracle APEX in a network that restricts internet access. In such a scenario, you must deploy a web application to provide the Oracle APEX image files.

Before you begin, you must identify the directory where you extracted the Oracle APEX distribution. In the following steps, this directory is referred to as `<apex directory>`. You then define a WebLogic web application which refers to this location and deploy that web application with a context-root value which is the default Oracle APEX image prefix: `i`. Create a directory which is the source for your web archive. In that directory, create a `WEB-INF` directory with the following files:

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/j2ee/dtds/web-app_2_3.dtd">

<web-app>
  <!-- This Web-App leverages the alternate doc-root functionality in WebLogic
to serve static content
      For WebLogic refer to the weblogic.xml file in this folder
  -->
</web-app>
```

weblogic.xml

```
<weblogic-web-app xmlns="http://www.bea.com/ns/weblogic/weblogic-web-app">
  <!-- This element specifies the context path the static resources are served
from -->
  <context-root>/i</context-root>
  <virtual-directory-mapping>
```



```
<!-- This element specifies the location on disk where the static resources
are located -->
<local-path>/scratch/apex/apex_21.2_en/images</local-path>
<url-pattern>*/</url-pattern>
</virtual-directory-mapping>
</weblogic-web-app>
```

Ensure that the local-path value in the weblogic.xml corresponds to your <apex directory>/images directory.

Where:

- <apex directory> is the directory location of Oracle APEX.

Create a Web Application with the web.xml and weblogic.xml files

```
jar cMf i.war WEB-INF/web.xml WEB-INF/weblogic.xml
```

The created images WAR does not contain the static resources; instead, it references the location where the static resources are stored. Therefore the static resources must be available at the specified path on the server where the WAR is deployed.

Deploy the i.war file to your WebLogic Server following the deployment steps that are applicable to your WebLogic Server environment. Once deployed, access the following URL to verify that the content is available:

```
http(s)://hostname:port/i/apex_version.txt
```

In response, the Oracle APEX version you have configured is displayed.

 **Tip:**

The i.war has a context-root with value /i. This is the default Oracle APEX image prefix. If you have configured Oracle APEX with the different image prefix value, then the context-root must be changed to that value.

6.1.2.3 Launching the Administration Server Console

To launch the Administration Server console:

1. Start an Administration Server.
2. Launch the WebLogic Administration Console by typing the following URL in your web browser:

```
http://<host>:<port>/console
```

Where:

- <host> is the DNS name or IP address of the Administration Server.
 - <port> is the port on which the Administration Server is listening for requests (port 7001 by default).
3. Enter your WebLogic Administrator username and password.

4. If your domain is in *Production* mode, click the **Lock & Edit** button on the left-pane below the submenu Change Center. If your domain is in *Development* mode, this button does not appear.

6.1.2.4 Deploying ORDS on Oracle WebLogic Server

Tip:

The Oracle REST Data Services files, `ords.war` and `i.war`, must be available before you start this task.

Note:

When you run ORDS in standalone mode using the `ORDS serve` command, you can specify the location of the configuration directory.

For deploying the `ords.war` in a supported container such as WebLogic Server, you must specify the configuration directory location. To do this, use one of the following approaches:

- The recommended approach is to generate a Web Application archive file with the configuration directory location specified for that `war` file.
- Set the `config.url` system property in the managed server so that it applies to all the ORDS deployments.
To generate the `war` file, use the `ords war` command.

To set the `config.url` system property, before starting WebLogic Server, run this command: `export JAVA_OPTIONS="-Dconfig.url=/scratch/my_ords_config"`

To deploy:

1. Go to the WebLogic Server Home Page. Below Domain Configuration, select **Deployments**.

The Summary of Deployments is displayed.

2. Click **Install**.
3. Specify the location of the `ords.war` file and click **Next**.

The `ords.war` file is located in the folder where you unzipped the Oracle REST Data Services ZIP file.

Tip:

WebLogic Server determines the context root from the file name of a WAR archive. If you need to keep backward compatibility, so that URLs are of the form `http://server/apex/...` rather than `http://server/ords/...`, then you must rename `ords.war` to `apex.war` before the deployment.

The Install Application assistant is displayed.

4. Select **Install this deployment as an application** and click **Next**.
5. Select the servers and/or clusters to which you want to deploy the application or module and click **Next**.

 **Tip:**

If you have not created additional Managed Servers or clusters, you do not see this assistant page.

6. In the Optional Settings, specify the following:
 - a. Name - Enter:
`ords`
 - b. Security - Select the following:
Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor
 - c. Source accessibility - Select:
Use the defaults defined by the deployment's targets
7. Click **Next**.
A summary page is displayed.
8. Under Additional configuration, select one of the following:
 - **Yes, take me to the deployment's configuration** - Displays the Configuration page.
 - **No I will review the configuration later** - Returns you to the Summary of Deployments page.
9. Review the summary of configuration settings that you have specified.
10. Click **Finish**.
11. Repeat the previous steps to deploy the `i.war` file.
In the optional settings, specify the following:
 - a. Name - Enter:
`i`
 - b. Security - Select:
Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor
 - c. Source Accessibility - Select:
Use the defaults defined by the deployment's targets
12. If your domain is in Production Mode, then on the Change Center click **Activate Changes**.

Related Topics

- [Installing and Configuring Oracle REST Data Services](#)
This chapter describes how to install, configure, and upgrade Oracle REST Data Services (ORDS).
- [Configuring Oracle APEX Images](#)

6.1.2.5 Configuring WebLogic to Handle HTTP Basic Challenges Correctly

By default WebLogic Server attempts to intercept all HTTP Basic Authentication challenges. This default behavior needs to be disabled for Oracle REST Data Services to function correctly. This is achieved by updating the `enforce-valid-basic-auth-credentials` flag. The WebLogic Server Administration Console does not display the `enforce-valid-basic-auth-credentials` setting. You can use WebLogic Scripting Tool (WLST) commands to check, and edit the value in a running server.

The following WLST commands display the domain settings:

```
connect('weblogic','weblogic','t3://localhost:7001')
cd('SecurityConfiguration')
cd('mydomain')
ls()
```

If the domain settings displayed, contains the following entry:

```
-r--  EnforceValidBasicAuthCredentials          true
```

Then you must set this entry to `false`.

To set the entry to `false`, use the WLST commands as follows:

```
connect('weblogic','weblogic','t3://localhost:7001')
edit()
startEdit()
cd('SecurityConfiguration')
cd('mydomain')
set('EnforceValidBasicAuthCredentials','false')
save()
activate()
disconnect()
exit()
```



Note:

WebLogic Server must be restarted for the new settings to take effect.

In the preceding example:

- `weblogic` is the WebLogic user having administrative privileges
- `weblogic` is the password
- `mydomain` is the domain
- The AdminServer is running on the `localhost` and on port `7001`

Related Topics

- [WebLogic Server Command Reference](#)

6.1.2.6 Verifying the State and Health of ords and i

In the Summary of Deployments, select the **Control** tab and verify that both the `ords` and `i` State are Active and the Health status is OK.

If `ords` and/or `i` are not Active, then enable them. In the Deployments table, select the check box next to `ords` and/or `i`. Click **Start** and select **Servicing all requests** to make them active.

6.1.3 Apache Tomcat

This section describes how to deploy Oracle REST Data Services on Apache Tomcat.

Topics:

- [About Apache Tomcat](#)
- [Configuring Oracle APEX Images](#)
- [Deploying ORDS on Apache Tomcat](#)

6.1.3.1 About Apache Tomcat

Tip:

This section assumes that you have completed the installation process and are familiar with Apache Tomcat. If you are unfamiliar with domains, servers, applications, security, users and roles, see your Apache Tomcat documentation.

You can download Apache Tomcat from:

See Also:

[Tomcat 8 Software Downloads](#)

6.1.3.2 Configuring Oracle APEX Images

This section is only applicable if you are using Oracle APEX. If you are not using Oracle APEX, then you can skip this section that provides details about configuring Oracle APEX Images.

If you are using Oracle APEX, then the Oracle APEX image files for that Oracle APEX version must also be accessible to the end user. It is recommended that you configure the Oracle APEX instance wherever applicable to use the resources from the Oracle Content Network.

The use of the Oracle CDN may not be feasible when using Oracle APEX in a network that restricts internet access. In such a scenario, you must deploy a web application to provide the Oracle APEX image files.

To configure Oracle APEX Images on Apache Tomcat:

- Copy the contents of the `<apex directory>/images` folder to `<Tomcat directory>/webapps/i/`.

Where:

- `<apex directory>` is the directory location of the Oracle APEX distribution.
- `<Tomcat directory>` is the folder where Apache Tomcat is installed.

6.1.3.3 Deploying ORDS on Apache Tomcat

Tip:

The Oracle REST Data Services file `ords.war` must be available before you start this task.

Note:

When you run ORDS in standalone mode using the `ORDS serve` command, you can specify the location of the configuration directory.

For deploying the `ords.war` in a supported container such as Apache Tomcat, you must specify the configuration directory location. To do this, use one of the following approaches:

- The recommended approach is to generate a Web Application archive file with the configuration directory location specified for that `war` file.
- Set the `config.url` system property in the managed server so that it applies to all the ORDS deployments.
To generate the `war` file, use the `ords war` command.

To set the `config.url` system property, before starting Apache Tomcat, run this command: `export JAVA_OPTS="-Dconfig.url=/scratch/my_ords_config"`

To deploy ORDS on Apache Tomcat:

1. Move the `ords.war` file into the `webapps` folder where Apache Tomcat is installed.

Tip:

Apache Tomcat determines the context root from the file name of a WAR archive. If you need to keep backward compatibility, so that URLs are of the form `http://server/apex/...` rather than `http://server/ords/...`, then you must rename `ords.war` to `apex.war` before moving it into to the `webapps` folder.

2. Access Oracle APEX by typing the following URL in your web browser:

```
http://<hostname>:<port>/ords/
```

Where:

- `<hostname>` is the name of the server where Apache Tomcat is running.
- `<port>` is the port number configured for Apache Tomcat application server.

Related Topics

- [Installing and Configuring Oracle REST Data Services](#)
This chapter describes how to install, configure, and upgrade Oracle REST Data Services (ORDS).
- [Configuring Oracle APEX Images](#)

6.1.4 Oracle Cloud Infrastructure

As a cloud user, you can deploy Oracle REST Data Services with high availability on Oracle Cloud Infrastructure.

Topics:

- [About Oracle Cloud Infrastructure \(OCI\)](#)
- [Configuring ORDS for High Availability on Oracle Cloud Infrastructure](#)
- [Advantages of Deploying ORDS with High Availability on Oracle Cloud Infrastructure](#)

**See Also:**

Reference Architecture: Deploying ORDS with High Availability

6.1.4.1 About Oracle Cloud Infrastructure

Oracle Cloud Infrastructure (OCI) is a set of complementary cloud services that enable you to build and run a wide range of applications and services in a highly available hosted environment. Oracle Cloud Infrastructure offers high-performance compute capabilities (as physical hardware instances) and storage capacity in a flexible overlay virtual network that is securely accessible from your on-premises network.

6.1.4.2 Configuring ORDS for High Availability on Oracle Cloud Infrastructure

You can choose to use the following configurations for high availability of ORDS on Oracle Cloud Infrastructure. Your requirements might differ from the configurations described in this section.

- When you create a VCN, determine the number of CIDR blocks required and the size of each block based on the number of resources that you plan to attach to subnets in the VCN. When you design the subnets, consider your traffic flow and security requirements.
- Use Oracle Cloud Guard to proactively monitor and maintain the security of your resources in Oracle Cloud Infrastructure.
- For production applications, the Oracle database instance should be adhering to Oracle Maximum Availability Architecture (MAA) deployment model in OCI.
- When using RAC with the Oracle Database, ensure that the database connection information used by ORDS is pointing to the SCAN listener and not an individual node.
- You can create load balancers with upper and lower bounds to help scale based on the number of requests coming in. It can be as small as 10mbps up to 8000mbps.

6.1.4.3 Advantages of Deploying ORDS with High Availability on Oracle Cloud Infrastructure

Following are the advantages of deploying ORDS on Oracle Cloud Infrastructure:

- Compute, load balancers, and Database Cloud Instances can all scale to handle increased load. Additional instances can be quickly created and added to the Load Balancer configuration with the compute or ORDS tier. For Exadata Cloud Service, the X8M platform can not only scale CPU, but also the nodes can be added to the RAC cluster to add additional computing power.
- Ensure that your subnet and NSG ingress/egress uses very granular rules. To get access to a compute or database tier, use Bastion as a Service, this ensures that only authorized users can access these instances and is also a much more secure method than exposing SSH ports to the public internet.
- For database deployments, adhere to the Oracle Maximum Availability Architecture (MAA) guide.
- Auto-scaling and scaling in general for each compute and database tier helps control costs and you need to pay only for what is being used with no excess or wasted CPU, memory, or instances. Using a flexible load balancer can also control costs.

6.2 Monitoring Oracle REST Data Services

Standard Java runtime environment diagnostic and monitoring tools are used to gain an insight on the health of an Monitoring Oracle REST Data Services (ORDS) instance running in Apache Tomcat, WebLogic Server, or a standalone mode. These tools track the memory and CPU usage, stuck threads, and other resources. ORDS provides additional insight through the ORDS instance API. The metrics available through the instance API makes it possible to check the status (valid or invalid) of the database pools and to gauge how the pools are being used. This helps in determining the actual load on the system and inform configuration changes in the future.

Topics:

- [Using the Instance API](#)
- [Using OpenTelemetry with Oracle REST Data Services](#)
- [ORDS CLI Commands for Managing Pools Using JMX](#)

6.2.1 Using the Instance API

The ORDS instance API service neither provides access to the database nor does it require the client to specify a database user for authentication. However, the ORDS instance returns information on the database pools. The instance API can be used as a basic health check service. To get a summary of the number of valid and invalid database pools, send a GET request to `/ords/_/instance-api/stable/status`. For example: `curl --user sysadmin:oracle http://<server>/ords/_/instance-api/stable/status`. This service returns a count of valid and invalid pools. It also returns links to additional information with more details on the database pools cache.

ORDS can be deployed as a single instance or in a cluster. In a cluster, you must address each instance directly to get the specific information about that specific instance as the database pool statistics for one instance may differ from the other instance. However, if the

load balancer routes to each instance in a round robin basis (as recommended), then every instance will have similar pool statistics.

6.2.1.1 Enabling the ORDS Instance API

This section explains how to enable the ORDS instance API.

To enable the ORDS instance API:

1. In `settings.xml` file or `pool.xml` file, add `instance.api.enabled` using the configuration command of `ords`:
 - For global `settings.xml`: `ords -config /path/to/config config set -global instance.api.enabled true`
 - For specific `pool.xml`: `ords -config /path/to/config config -db-pool mypool set instance.api.enabled true`
2. Restart ORDS.

6.2.1.2 Authorization for Using the ORDS Instance API

The System Administrator role is required to use the ORDS instance API. For production environments, it is recommended that a user with this role is configured through the mid-tier.

6.2.1.3 API Document

An OpenAPI description of the ORDS instance API services is available at `http://<server>/ords/_/instance-api/stable/metadata-catalog/openapi.json`.

6.2.2 Using OpenTelemetry with Oracle REST Data Services

This section describes how to configure ORDS to use OpenTelemetry.

ORDS can be configured to provide telemetry data (metrics, logs, and traces) through OpenTelemetry. OpenTelemetry, also known as OTEL in short, is a vendor-neutral open-source observability framework for instrumenting, generating, collecting, and exporting telemetry data to assist in the analysis and tuning of software systems. As an industry standard it is natively supported by a number of vendors.

Configure ORDS to export telemetry data to an OpenTelemetry compliant collector

Perform the following steps to run ORDS to exports telemetry data to an OpenTelemetry collector:

1. Download the latest OpenTelemetry Java Agent.
2. Configure OpenTelemetry Java Agent

See Also:

- [Download OpenTelemetry Java agent](#)
- [Configure OpenTelemetry Java Agent](#)

3. Add `-javaagent` option to `JDK_JAVA_OPTIONS` for a standalone server or `CATALINA_OPTS` for Tomcat.

The following example configures OpenTelemetry agent to export telemetry data to an OpenTelemetry Collector, running on host `otel-collector-host` and running ORDS in a standalone mode:

```
export OTEL_SERVICE_NAME=ords
export OTEL_METRICS_EXPORTER=none
export OTEL_TRACES_EXPORTER=otlp
export OTEL_EXPORTER_OTLP_ENDPOINT=http://otel-collector-host:4317
export JDK_JAVA_OPTIONS=-javaagent:/path/to/opentelemetry-javaagent.jar
ords --config /path/to/config serve
```

In the preceding example, the Java agent is specified using the Java standard `JDK_JAVA_OPTIONS` environment variable.

When you deploy ORDS on Apache Tomcat or Oracle WebLogic Server, the same Java Agent and environment variables can be specified. See Apache Tomcat or Oracle WebLogic server documentation for the most appropriate approach to use for that platform.

WARNING:

`JDK_JAVA_OPTIONS` affects all the Java programs. Configure this option before the `ords` command, or use `APP_VM_OPTS` to configure the Javaagent for ORDS.

For example:

```
export APP_VM_OPTS=-javaagent:/path/to/opentelemetry-javaagent.jar
```

Configuring OpenTelemetry Collector to export telemetry data to Oracle APM

Oracle APM provides a comprehensive set of features to monitor the applications. It supports OpenTelemetry tracing.

To export telemetry traces to APM, perform the following steps:

1. Create an APM Domain in OCI.
2. Record the `<Data upload endpoint>` of the APM domain.
3. Create and record the `<Private Data Key>` of the APM domain.
4. Install `otelcol`.
Refer to [OpenTelemetry documentation](#) for the installation instructions
5. Configure the Zipkin exporter in the `otelcol` configuration file. For example: `/etc/otelcol/config.yaml` on Linux. Replace `<Data Upload Endpoint>` and `<Private Data Key>` with the actual endpoints and keys.

Example code snippet:

Language: `yml`

```
extensions:
  health_check:
  pprof:
    endpoint: 0.0.0.0:1777
```

```
zpages:
  endpoint: 0.0.0.0:55679

receivers:
  otlp:
    protocols:
      grpc:
      http:

processors:
  batch:

exporters:
  logging:
    verbosity: detailed

zipkin:
  endpoint: "<Data Upload Endpoint>/20200101/observations/private-span?
dataFormat=zipkin&dataFormatVersion=2&dataKey=<Private Data Key>"

service:
  pipelines:
    traces:
      receivers: [otlp]
      processors: [batch]
      exporters: [logging, zipkin]

    metrics:
      receivers: [otlp]
      processors: [batch]
      exporters: [logging]

extensions: [health_check, pprof, zpages]
```

6. Run the `otelcol` service with the updated configuration. With this configuration, OpenTelemetry Collector (`otelcol`) exports telemetry traces to Oracle APM using the Zipkin exporter with the specified `<Data Upload Endpoint>` and `<Private Data Key>`

 **Note:**

The `OTEL_XXXX` environment variables are described in detail in OpenTelemetry documentation.

 **See Also:**

- [OpenTelemetry Documentation](#)
- [Apache Tomcat](#)
- [Oracle WebLogic Server](#)

6.2.3 ORDS CLI Commands for Managing Pools Using JMX

This section describes the ORDS CLI commands to list or destroy the connection pools through JMX based pool management.

ORDS pool command uses the Java Attach API to connect to the local Java process running ORDS and then lists or destroys the Universal Connection Pools (UCP) through JMX based pool management.

See Also:

- [JMX based pool management](#)
- [Attach API](#)
- [Local Monitoring and Management Using JConsole](#)

In order to use the `ords pool` command:

- ORDS must have JMX UCP enabled. It is enabled by default.
- You must specify the Oracle REST Data Services instance process ID.

The `ords pool` command can be used to list the pools or destroy the pools without the need to restart the service.

- **List the pools in a running ORDS instance:**

Note:

Before running the `list` command, identify the process ID of the ORDS instance. To identify the process ID of the ORDS instance, use the command `jps -ml` to list the running Java processes on the local host.

Run the following command to list the available pools, specify the identified process ID:

```
ords pool --pid 3459 list
```

In response, the list of available pools is displayed in the console.

- **Destroy a pool in a running ORDS instance:**

To destroy the specified pool name:

Run the following destroy command specifying the process ID and the pool name enclosed in double quotes:

```
ords pool --pid 3459 destroy "|pdb2|lo|-2023-05-24T21-28-03.076379Z"
```

A message is displayed indicating that the pool has been destroyed

The pool `|pdb2|lo|-2023-05-24T21-28-03.076379Z` was destroyed.

7

Using the Multitenant Architecture with Oracle REST Data Services

This section outlines installing, configuring, upgrading and uninstalling Oracle REST Data Services in a multitenant container database.

- [Setting Up ORDS in a CDB Environment](#)
- [Setting Up ORDS in an Application Container](#)

Once ORDS metadata is saved in the database, it is necessary to provide a runtime configuration setting for ORDS to efficiently map requests to the corresponding pluggable databases.



See Also:

[Making All PDBs Addressable by Oracle REST Data Services \(Pluggable Mapping\)](#)

7.1 Setting Up ORDS in a CDB Environment

This section describes how to setup Oracle REST Data Services (ORDS) into a multitenant container database (CDB) environment.

Oracle Database 12c Release 1 (12.1) introduced the multitenant architecture. This database architecture has a multitenant container database (CDB) that includes a root container, `CDB$ROOT`, a seed database, `PDB$SEED`, and multiple pluggable databases (PDBs). A PDB appears to users and applications as if it were a non-CDB. Each PDB is equivalent to a separate database instance in Oracle Database Release 11g.

The root container, `CDB$ROOT`, holds common objects that are accessible to every PDB utilizing metadata links or object links. The seed database, `PDB$SEED`, is used when you create a new PDB to seed the new pluggable database. The key benefit of the Oracle Database 12c multitenant architecture is that the database resources, such as CPU and memory, can be shared across all of the PDBs. This architecture also enables many databases to be treated as one for tasks such as upgrades or patches, and backups.

The installation process when you have multiple releases is described in the following section:

- [Installation Enabling Multiple Releases](#)

Preinstallation Tasks for Oracle REST Data Services CDB Installation

- Ensure that the PDBs are open (not mounted/closed) in read/write mode (except for `PDB$SEED`, which remains in read-only mode). For more information, see Oracle Multitenant Administrator's Guide
- Ensure that the default and temporary tablespaces to be used by the `ORDS_METADATA` schema and the `ORDS_PUBLIC_USER` user exist and that you know the tablespace names. The installation procedure creates those users, but it does not create the tablespaces.

- Ensure that when a common user is used for the PL/SQL Gateway, the user has also been explicitly created in the PDB\$SEED container otherwise, the installer will not be able to configure the PL/SQL Gateway correctly.

 **Note:**

ORDS_METADATA and ORDS_PUBLIC_USER are installed in the seed container, and the default and temporary tablespaces exist in PDB\$SEED. If these tablespaces do not already exist, then you must create the tablespaces in PDB\$SEED. For more information, see Oracle Multitenant Administrator's Guide

7.1.1 Installation Enabling Multiple Releases

This section describes the installation process when you have multiple releases of Oracle REST Data Services and patch sets in the PDBs in a multitenant environment.

When Oracle REST Data Services is installed into a CDB, the proxy user, Oracle REST Data Services public user (ORDS_PUBLIC_USER) is installed in the root container and is a common user. The ORDS_METADATA schema is a local user that contains the metadata for Oracle REST Data Services. Both the ORDS_METADATA schema and the ORDS_PUBLIC_USER are installed in the seed container (PDB\$SEED) and all of the pluggable databases.

Since the ORDS_METADATA is installed as a local user, this provides you the flexibility of installing multiple Oracle REST Data Services releases in the pluggable databases.

7.1.1.1 Command Line Installation

You must provide the SYS AS SYSDBA credentials in the Root (CDB\$ROOT) container to perform the installation.

7.1.1.2 Advanced Installation

This section describes the advanced installation prompts for installing Oracle REST Data Services into a CDB to enable multiple Oracle REST Data Services releases.

To install Oracle REST Data Services into a CDB to enable multiple Oracle REST Data Services releases, perform the following steps:

1. Navigate to the folder where you unzipped the Oracle REST Data Services installation kit.
2. Enter the following command:

```
ords --config <CONFIG_FOLDER> install
```

3. Choose the installation type required

```
Enter a number to select the type of installation
  [1] Install or upgrade ORDS in the database only
  [2] Create or update a database pool and install/upgrade ORDS in the
database
  [3] Create or update a database pool only
Choose [2]:
```

- Accept or enter 2 (the default) to update the configuration and install ORDS REST Data Services into the CDB and all of its PDBs.

```
Enter a number to select the database connection type to use
  [1] Basic (host name, port, service name)
  [2] TNS (TNS alias, TNS directory)
  [3] Custom database URL
Choose [1]:
```

- Accept or enter 1 (the default) to enter the connection information to the CDB

```
Enter the database host name [localhost]: Enter the database listen port [1521]: Enter the
database service name [orcl]: Provide database user name with administrator privileges.
Enter the administrator username: SYS Enter the database password for SYS AS
SYSDBA: Connecting to database user: SYS AS SYSDBA url: jdbc:oracle:thin:@//
localhost:1521/orcl Retrieving information... Your database connection is to a CDB. ORDS
common user ORDS_PUBLIC_USER will be created in the CDB. ORDS schema will be
installed in the PDBs. Root CDB$ROOT - create ORDS common user PDB PDB$SEED -
install ORDS xx.x.x.xxx.xxxx (mode is READ ONLY, open for READ/WRITE) PDB
PDBName1 - install ORDS xx.x.x.xxx.xxxx PDB PDBName1 - configure PL/SQL gateway
user APEX_PUBLIC_USER in ORDS version xx.x.x.xxx.xxxx Install ORDS in the database
[1] Yes [2] No Choose [1]:
```

```
Enter 1 if you want to install ORDS or 2 to skip this step [1]:
```

- Press enter to continue with the installation.
- When prompted, enter additional information as needed.



Note:

To use the pluggable mapping feature, see [Making All PDBs Addressable by Oracle REST Data Services \(Pluggable Mapping\)](#) for more information.

7.1.1.3 Silent Installation

Silent installation reads the properties from the Oracle REST Data Services parameter file.

To perform a silent installation, enter the following command:

```
ords --config /path/to/config install <OPTIONS>
```

7.1.2 Upgrading Oracle REST Data Services in a CDB Environment

When you use a new release of Oracle REST Data Services, upgrading its schema in the CDB and its pluggable databases (PDBs) will occur automatically when you perform a simple or advanced installation.

For example:

```
ords install --interactive --log-folder <LOG_FOLDER>
```

If Oracle REST Data Services is already installed or upgraded, a message displays the Oracle REST Data Services schema version, and you will not be prompted for information.

7.1.2.1 Migrating Oracle REST Data Services in the CDB to Enable Multiple Releases

This section describes how to migrate Oracle REST Data Services in the CDB to enable multiple releases.

Starting with release 18.2.0 and later, if you have an Oracle REST Data Services schema and `ORDS_METADATA` that is installed in the `CDB$ROOT` container, then during upgrade it will migrate the common `ORDS_METADATA` schema to your PDBs as a local schema. Oracle database 12.1.0.2 and later releases support this change.

7.1.3 Uninstalling Oracle REST Data Services in a CDB Environment

To uninstall Oracle REST Data Services from a CDB, use the `uninstall` command.

For example:

```
ords uninstall --interactive --log-folder <LOG_FOLDER>
```

Oracle REST Data Services will be removed from the CDB and its pluggable databases (PDBs).

7.2 Setting Up ORDS in an Application Container

This section describes how to setup Oracle REST Data Services in an application container.

Starting with ORDS release 20.2.1, Oracle REST Data Services can be installed or upgraded into an application container using the ORDS SQL scripts provided in the `ords.version.number.zip` file.

An application container consists of an application root where the application is defined and one or more PDBs that share the metadata about the application from the application root. You can have multiple application containers within a CDB and each container can have different versions of Oracle REST Data Services. Installing or upgrading Oracle REST Data Services in an application container is done against the application root container. When an application PDB wants to use the upgraded version, it must synchronize with the application root. Oracle REST Data Services continues to run in the application PDB with the existing version until the application PDB synchronizes with the application root.

Topics:

- [Prerequisites for Creating ORDS in an Application Container](#)
- [Installing ORDS in the Application Root Container](#)
- [Creating an Application Seed](#)
- [Creating an Application PDB from the Application Seed](#)
- [ORDS Configuration Files Setup](#)
- [Running ORDS](#)
- [Upgrading ORDS in the Application Container](#)
- [Uninstalling ORDS from the Application Container](#)
- [Verifying ORDS in the Application Container](#)

7.2.1 Prerequisites for Creating ORDS in an Application Container

This section describes the prerequisites for installing ORDS in an application container.

Following prerequisites must be met before you install ORDS in an application container:

- Download ORDS version 20.2.1 or later from Oracle REST Data Services Downloads.
- Extract the ORDS SQL scripts.
- To obtain the ORDS SQL scripts, execute the following commands:

```
unzip ords.version.number.zip ords.war
unzip ords.war 'WEB-INF/lib/ords-installer-*.jar'
unzip 'WEB-INF/lib/ords-installer-*.jar' 'db/*'
mv db scripts
```

The ORDS SQL scripts are located in the scripts folder. The scripts folder contains the subdirectories for the install, upgrade, validate, and uninstall SQL scripts. You can run these SQL scripts using SQLcl, SQL*Plus, or SQL Developer.

7.2.1.1 Creating an Application Root Container

This section describes how to create an application root container.

To create an application root container:

1. Ensure that the current container is in `CDB$ROOT`.
2. Use the `AS APPLICATION CONTAINER` clause of the `CREATE PLUGGABLE DATABASE` statement to create an application container.
3. Open the application container.

Example:

```
CREATE PLUGGABLE DATABASE ords_app_root1 AS APPLICATION CONTAINER ADMIN USER
admin IDENTIFIED BY <admin_password>

FILE_NAME_CONVERT=('pdbseed', 'ords_app_root1');

ALTER PLUGGABLE DATABASE ords_app_root1 OPEN;
```

Note:

`ords_app_root1` and the `admin` user in the preceding example can be any valid Oracle identifier.

If Oracle managed files is enabled in the CDB or the `PDB_FILE_NAME_CONVERT` initialization parameter is set, then omit the `FILE_NAME_CONVERT` clause.

The ORDS users, namely `ORDS_PUBLIC_USER` and `ORDS_METADATA`, must not exist in the seed (for example, `pdbseed`) or cloned `pdb`.

**See Also:**

[Creating an Application Container](#)

7.2.2 Installing ORDS in the Application Root Container

This section describes how to install ORDS in the application root container.

To install ORDS in the application root container, perform the following steps:

1. Connect to the application root container.
2. Run `/path/to/scripts/install/core/ords_app_con_install.sql` command using the following parameters:
 - Log folder (must include the forward slash at the end)
 - Default tablespace for ORDS schema
 - Temporary tablespace for ORDS schema
 - Default tablespace for ORDS proxy user
 - Temporary tablespace for ORDS proxy user
 - ORDS proxy user password
 - Scripts path (requires the fully qualified path to the ORDS scripts)

**Note:**

The tablespaces must already exist in the database.

```
ALTER SESSION SET CONTAINER = ords_app_root1;
```

```
@/path/to/scripts/install/core/ords_app_con_install.sql /path/to/logs/ SYSAUX  
TEMP SYSAUX TEMP P033w0r6! /path/to/scripts
```

Where:

The `ords_app_con_install.sql` creates an application named ORDS and assigns the application version to the ORDS product version. The product version format is Year.Quarter.Patch.rJulianDay24HRMM (for example, 20.2.1.r2121800).

The preceding script installs ORDS and creates the following:

- The ORDS schema, `ORDS_METADATA`
- The ORDS proxy user, `ORDS_PUBLIC_USER` and
- The related database objects in the application container

**See Also:**

[Verifying ORDS in the Application Container](#)

7.2.3 Creating an Application Seed

This section describes how to create an application seed.

An application seed is used to provision application PDBs with the application root's applications pre-installed.

To create an application seed:

1. Ensure that the current container is in the `CDB$ROOT`.
2. Alter session and set container to the application root.
3. Use the `AS SEED` clause of the `CREATE PLUGGABLE DATABASE` statement to create an application seed.
4. Sync the ORDS application with the application seed.
5. Compile invalid objects.
6. Open the application seed in a read only mode.

Note:

`ords_app_root1` and the `admin` user in the following example can be any valid Oracle identifier.

If Oracle managed files is enabled in the CDB or the `PDB_FILE_NAME_CONVERT` initialization parameter is set, then omit the `FILE_NAME_CONVERT` clause.

```
ALTER SESSION SET CONTAINER = ords_app_root1;
CREATE PLUGGABLE DATABASE AS SEED ADMIN USER admin IDENTIFIED BY
<admin_password>
FILE_NAME_CONVERT=('pdbseed', 'ords_app_root1_seed');
ALTER PLUGGABLE DATABASE ords_app_root1$seed open;
ALTER SESSION SET CONTAINER = ords_app_root1$seed;
ALTER PLUGGABLE DATABASE application ORDS sync;
begin
  sys.dbms_utility.compile_schema('ORDS_METADATA', FALSE);
end;
/
ALTER PLUGGABLE DATABASE ords_app_root1$seed close immediate;
ALTER PLUGGABLE DATABASE ords_app_root1$seed open read only;
```

See Also:

[Creating an Application Container](#)

7.2.4 Creating an Application PDB from the Application Seed

This section describes how to create an application PDB that is seeded from the application seed.

An application PDB is created by issuing the `CREATE PLUGGABLE DATABASE` statement from the application root.

To create an application PDB from the application seed:

1. Ensure that the current container is in `CDB$ROOT`.
2. Alter session and set the container to the application root.
3. Use the `CREATE PLUGGABLE DATABASE` command to create a PDB from the application seed.

Note:

`ords_app_pdb1` and the `admin` user in the following example can be any valid Oracle identifier.
If Oracle managed files is enabled in the CDB or the `PDB_FILE_NAME_CONVERT` initialization parameter is set, then omit the `FILE_NAME_CONVERT` clause.

```
ALTER SESSION SET CONTAINER=ords_app_root1;
CREATE PLUGGABLE DATABASE ords_app_pdb1 ADMIN USER admin IDENTIFIED BY <admin
password>
FILE_NAME_CONVERT=('ords_app_root1_seed','ords_app_pdb1');
ALTER PLUGGABLE DATABASE ords_app_pdb1 OPEN;
ALTER SESSION SET CONTAINER = ords_app_pdb1;
select app_name, app_version, app_status from dba_applications where app_name
= 'ORDS';
```

APP_NAME	APP_VERSION	APP_STATUS
ORDS	20.2.1.r2121800	NORMAL

See Also:

Creating an Application Container

7.2.5 Configuring the PLSQL Gateway for APEX

This section describes how to configure the PL/SQL Gateway to work with APEX, if required. The `APEX_PUBLIC_USER` must be configured to proxy to the `ORDS_PUBLIC_USER` to support ORDS RESTful Services from within APEX.

To configure the PL/SQL Gateway:

- Ensure that the current container is in `CDB$ROOT`.

- Alter session and set the container to the application pdb, example: `ords_app_pdb1`.

```
ALTER SESSION SET CONTAINER = ords_app_pdb1;
begin
  ords_admin.config_plsql_gateway(
    p_runtime_user      => 'ORDS_PUBLIC_USER',
    p_plsql_gateway_user => 'APEX_PUBLIC_USER',
    p_comments          => 'APEX proxied to ORDS');
  commit;
end;
/
```

7.2.6 ORDS Configuration Files Setup

This section describes how to setup the ORDS configuration files:

Topics:

- [Specifying the ORDS Configuration Folder](#)
- [Creating the ORDS Configuration Files for the Application Root Container](#)

7.2.6.1 Specifying the ORDS Configuration Folder

This section describes how to specify the ORDS configuration folder.

You must provide your configuration folder prior to creating your configuration.



See Also:

[Setting Up the Configuration Folder Structure](#)

7.2.6.2 Creating the ORDS Configuration Files for the Application Root Container

This section describes how to create the ORDS configuration files for the application root container.

To create the ORDS configuration files for the application root container, execute the following setup command to create the configuration files:

```
ords --config <configuration folder> install --interactive --config-only --
prompt-password --log-folder <Log Folder>
```

Where the `--config-only` option must be specified to create the configuration files. When prompted for the service name, specify the application root servicename.

Example 7-1 Creating ORDS Configuration

This example assumes that ORDS configuration does not exist in your configuration folder. This example is for a Basic Connection.

 **Note:**

- If Oracle APEX is installed in your database, then a prompt is displayed for the APEX images location.
Example: Enter the APEX static resources location: /path/to/apex/images
- If you are not using APEX, then press enter.
- Oracle recommends to use HTTPS protocol instead of HTTP.

Oracle REST Data Services - Interactive Install

```
Enter a number to select the type of installation
  [1] Install or upgrade ORDS in the database only
  [2] Create or update a database pool and install/upgrade ORDS in the
database
  [3] Create or update a database pool only
Choose [3]:
Enter a number to select the database connection type to use
  [1] Basic (host name, port, service name)
  [2] TNS (TNS alias, TNS directory)
  [3] Custom database URL
Choose [1]:
Enter the database host name [localhost]:
Enter the database listen port [1521]:
Enter the database service name [orcl]: ords_app_root1
Enter the database password for ORDS_PUBLIC_USER:
Confirm password:
Enter a number to select additional feature(s) to enable:
  [1] Database Actions (Enables all features)
  [2] REST Enabled SQL and Database API
  [3] REST Enabled SQL
  [4] Database API
  [5] None
Choose [1]:
Enter a number to configure ORDS for standalone mode
  [1] Configure ORDS for standalone mode
  [2] Skip
Choose [1]:
Enter a number to use HTTP or HTTPS protocol
  [1] HTTP
  [2] HTTPS
Choose [1]: 2
Enter the HTTPS port [8443]:
Enter a number to select the certificate type
  [1] Use self-signed certificate (generates automatically)
  [2] Use my SSL certificate (requires SSL certificate and SSL certificate
private key)
Choose [1]: 2
Enter the path for the SSL Certificate: /path/to/<SSL certificate>
Enter the path for the SSL Certificates private key: /path/to/<SSL
```

```
Certificate private key>  
Enter the APEX static resources location: /path/to/apex/images
```

7.2.7 Running ORDS

This section lists the different methods you can use to run ORDS after installing ORDS in the application container.

Once you install ORDS in the application container and create the ORDS configuration files, run ORDS using one of the following methods:

- Standalone Mode
- Deploy on Oracle WebLogic Server
- Deploy Oracle REST Data Services on Apache Tomcat



See Also:

[Deploying Oracle REST Data Services](#)

7.2.8 Upgrading ORDS in the Application Container

This section describes how to upgrade ORDS in the application container.

You can upgrade an application in an application container. These operations are performed in the application root. The application container propagates the application changes to the application PDBs when the application PDBs synchronize with the application in the application root.

Prerequisites:

- ORDS must already be installed in the application container.
- Upgrading ORDS from an earlier release to a new release (for example, ORDS release 20.2.x.x to 20.3.x.x).

To upgrade ORDS in the application root:

1. In SQLcl or SQL*Plus, connect to the application root.
2. Run `/path/to/scripts/upgrade/ords_app_con_upgrade.sql` with the following parameters:
 - Log folder (must include the forward slash at the end)
 - Scripts path (requires the fully qualified path to the ORDS scripts)



Note:

The `ords_app_con_upgrade.sql` script upgrades ORDS in the application root container to the release that you are using. For example, if the ORDS application version is 20.2.1.r2121800, and the ORDS upgrade script is 20.3.0.r2601900, then the script upgrades ORDS to release 20.3.0.r2601900 in the application root container.

To synchronize the ORDS application in an application PDB with the upgrade changes in the application root:

1. In SQLcl or SQL*Plus, ensure that the current container is the application PDB.
2. Run the `ALTER PLUGGABLE DATABASE APPLICATION` statement specifying the ORDS application with the `SYNC` clause.

```
ALTER SESSION SET CONTAINER = ords_app_pdb1;
```

```
ALTER PLUGGABLE DATABASE APPLICATION ORDS SYNC;
```



See Also:

[Verifying ORDS in the Application Container](#)

7.2.9 Uninstalling ORDS from the Application Container

This section describes how to uninstall ORDS from the application container.

You can uninstall an application from an application container. These operations are performed in the application root. The application container propagates the application changes to the application PDBs when the application PDBs synchronize with the application in the application root.

To uninstall ORDS from the application root:

1. In SQLcl or SQL*Plus, connect to the application root.
2. Run `/path/to/scripts/uninstall/core/ords_app_con_uninstall.sql` with the following parameters:
 - Log folder (must include the forward slash at the end)
 - Scripts path (requires the fully qualified path to the ORDS scripts)

```
ALTER SESSION SET CONTAINER = ords_app_root1;
```

```
@/path/to/scripts/uninstall/core/ords_app_con_uninstall.sql /path/to/logs/ /  
path/to/scripts
```

To synchronize the application PDB to uninstall the ORDS application:

1. In SQLcl or SQL*Plus, ensure that the current container is the application PDB.
2. Run the `ALTER PLUGGABLE DATABASE APPLICATION` statement specifying the ORDS application with the `SYNC` clause.

For example:

```
ALTER SESSION SET CONTAINER = ords_app_pdb1;
```

```
ALTER PLUGGABLE DATABASE APPLICATION ORDS SYNC;
```



See Also:

[Verifying ORDS in the Application Container](#)

7.2.10 Verifying ORDS in the Application Container

This section describes how to verify ORDS in the application container.

To verify the ORDS for install, upgrade, validate, and uninstall in the application container:

- Manually inspect the following log files for any errors:
 - Install - ordsinstall_<timestamp>.log
 - Upgrade - ordsupgrade_<timestamp>.log
 - Uninstall - ordsuninstall_<timestamp>.log
- Query `dba_applications` to verify if the ORDS application exists and its application version is the same as the ORDS product version.

```
SQL> select app_name, app_version, app_status from dba_applications where
app_name = 'ORDS';
```

APP_NAME	APP_VERSION	APP_STATUS
ORDS	20.2.1.r2121800	NORMAL

- Query `dba_app_errors` to check for any errors:


```
SQL> select app_name, app_statement, errornum, errormsg from dba_app_errors
where app_name = 'ORDS';
```

no rows selected

If you are uninstalling ORDS from the application container, the `APP_STATUS` contains the value `UNINSTALLED`.

7.3 Making All PDBs Addressable by Oracle REST Data Services (Pluggable Mapping)

Pluggable mapping refers to the ability to make all PDBs in a CDB root or in an application root container addressable by Oracle REST Data Services. To use this feature, follow the instructions described in this topic.

If the Oracle REST Data Services configuration file includes the `db.serviceNameSuffix` parameter, this indicates that the Oracle REST Data Services pool points to a CDB root or an application root, and that the PDBs connected to that CDB root or an application root should be made addressable by Oracle REST Data Services. For Pluggable mapping to work, you must specify this parameter setting. You can specify the `db.serviceNameSuffix` parameter at a pool-specific level or at a global level. If you set the parameter to a global level, then you can apply it to all the database pool configurations that define a CDB connection. Also, the Database Actions landing page (`<server-name>:<port-number>/ords/sql-developer`) displays a prompt for the PDB name, if you set the parameter to a global level.

The value of the `db.serviceNameSuffix` parameter must correspond to the value of the `DB_DOMAIN` database initialization parameter, based on the following rules:

- If the `DB_DOMAIN` value is blank, then the `db.serviceNameSuffix` parameter must also be an empty string.

- If the `DB_DOMAIN` value is not blank, then the `db.serviceNameSuffix` parameter must have a value starting with a period (.).

To set the value of the `db.serviceNameSuffix` parameter:

1. In SQLcl or SQL*Plus, connect to the root as a user with SYSDBA privileges.
2. Check the value of the `DB_DOMAIN` database initialization parameter.

```
SQL> show parameter DB_DOMAIN
```

3. Exit SQLcl or SQL*Plus.

```
SQL> exit
```

4. If the `db.serviceNameSuffix` parameter value is not defined, enter a command in the following format to add an entry to the configuration file:

```
ords --config /path/to/config config --db-pool pool-name set "db.serviceNameSuffix" ".value-of-DB_DOMAIN"
```

Where `pool-name` is one of the following:

- `default` for the default and migrated pool
- `<custom_pool>` as specified during ORDS install

Example 1: You want to make PDBs in a CDB root or an application root addressable globally. Specify global settings by entering the following command:

```
ords --config /path/to/config config set --global "db.serviceNameSuffix" ".value-of-DB_DOMAIN"
```

Note:

The approach shown in Example 1, that is, setting the property for all the pools through the `global/settings.xml` file, works the best for most use cases.

Example 2: You want to make PDBs in a CDB root or an application root addressable for your PL/SQL Gateway, and your pool name is `my_pool`. Enter the following command:

```
ords --config /path/to/conf config --db-pool my_pool set "db.serviceNameSuffix" ".example.com"
```

For example, if the database pointed to by apex has a `DB_DOMAIN` value of `example.com` and contains the two PDBs `pdb1.example.com` and `pdb2.example.com`, the first PDB will be mapped to URLs whose path starts with `/ords/pdb1/`, and the second PDB will be mapped to URLs whose path starts with `/ords/pdb2/`.

Example 3: You want to make PDBs in a CDB root or an application root addressable globally and the `DB_DOMAIN` is an empty value. Enter the following command to specify `db.serviceNameSuffix` parameter globally:

```
ords --config /path/to/config config set --global "db.serviceNameSuffix" ""
```

8

Installing and Configuring Customer Managed ORDS on Autonomous Database

This section explains how to install and configure Customer Managed Oracle REST Data Services (ORDS) on Autonomous Database.

Topics:

- [About Customer Managed Oracle REST Data Services on Autonomous Database](#)
- [Preinstallation Tasks](#)
- [ORDS Command-Line Interface for Customer Managed ORDS](#)

8.1 About Customer Managed Oracle REST Data Services on Autonomous Database

When you provision an Autonomous Database instance, by default Oracle REST Data Services (ORDS) is preconfigured and available for the instance. With the default ORDS, Oracle performs any required configuration, patching, and maintenance. Additionally, you can also configure Autonomous Database to use ORDS running in a customer managed environment.

When you use the default ORDS on Autonomous Database, you cannot modify any of the ORDS configuration options. For example, with the default configuration, the JDBC connection pools have a maximum of 100 connections and the connections for ORDS are preconfigured to use the `LOW` database service. Use a customer managed environment if you want manual control of the configuration and management of Oracle REST Data Services. For example, use this option when your applications require larger connection pools or if you need more control over the ORDS configuration options.

When ORDS runs in a customer managed environment, you are responsible for configuration, patching, and maintenance of ORDS in the customer managed environment. After you configure Autonomous Database to use your customer managed ORDS in addition to the existing autonomously managed ORDS, you can route ORDS HTTPS traffic through your environment. The default Autonomous Database web server and ORDS are still running and ORDS traffic goes to the ORDS running in the customer managed environment. This provides an additional and alternative HTTPS solution for Autonomous Database.

Installing and configuring a customer managed environment for ORDS allows you to run ORDS with configuration options that are not possible using the default Oracle managed ORDS available with Autonomous Database.

Installing and configuring a customer managed environment for ORDS is only supported with Autonomous Database.

 **Note:**

- A version of Oracle REST Data Services within one major release of what is running in Autonomous Database (Serverless or Dedicated) is required to use a customer managed environment for ORDS with a version closest to that release. For example:

If your Autonomous Database is currently hosting ORDS version 22.4, then you must be running the latest possible ORDS version with customer managed environment. Maintain the latest version of ORDS as close as possible, for example 23.1 or 22.3 for the best results.

 **Note:**

The latest version of ORDS can be found in the OCI YUM repository and also at [ORDS Downloadable Zip File](#) location.

8.2 Preinstallation Tasks

This section describes the preinstallation tasks.

Before you begin:

- Download the wallet from your Oracle Autonomous Database instance.
- If you are using ORDS with Oracle APEX, then you are required to setup the Oracle APEX static resources.

8.2.1 Downloading Wallet

You need to configure ORDS to connect to the Autonomous Database. With Oracle REST Data Services (ORDS) running in a customer managed environment, you need to obtain the Autonomous Database wallet on the system that runs the customer managed ORDS.

- To download the wallet for the Autonomous Database instance, see [Download Client Credentials \(Wallets\)](#) for the detailed steps.

8.2.2 Oracle APEX Static Resources

This section describes how to set up the APEX static resources.

If you are using ORDS and APEX, then setting up the APEX static resources is mandatory. You can setup the APEX static resources by using the Oracle Content Delivery Network (CDN), or downloading APEX and copying the APEX images folder to your environment.

1. Oracle recommends using the Oracle Content Delivery Network to setup the APEX static resources. See [Customer Managed ORDS on Autonomous Database](#) for detailed instructions.

 **Note:**

You only need to setup APEX static resources once using CDN, then APEX automatically upgrades this for you in the Autonomous Database.

2. Download APEX and configure APEX static resources.
 - Download APEX from the location Oracle APEX Downloads
 - Copy the images directory. See Copying the Images Directory

 **Note:**

You must download, maintain, and upgrade the APEX static resources and ensure that the APEX version that you are using is consistent with the APEX version on the Autonomous Database.

 **See Also:**

Control Oracle APEX Upgrades

8.3 ORDS Command-Line Interface for Customer Managed ORDS

The ORDS Command-Line Interface (CLI) provides the interactive and silent command `install adb` to automate configuring a Customer Managed ORDS. This includes creating the ORDS configuration in your environment. If you want to use Autonomous Database with Oracle REST Data Services (ORDS) running in a customer managed environment, execute this command. This creates an ORDS runtime database user. ORDS can connect, and provide the privileges to that runtime user. In addition, it creates and provides privileges to the PL/SQL gateway database user used for APEX, PL/SQL Gateway and OWA, and allow connections through the runtime user. The runtime database user and gateway database user are created in the Autonomous Database.

8.3.1 Interactive Install for ADB Command Line Interface

Use the ORDS Command-Line Interface (CLI) to interactively prompt you for the following information to setup your Customer Managed ORDS.

- Wallet Path
- Net Service Name from `tnsnames.ora` contained in the wallet zip file
- Administrator user
- Runtime database user
- PL/SQL gateway user
- Additional Database Features
- Standalone options

Examples:

- `ords install adb`
- `ords install adb --interactive [OPTIONS]`
- `ords install adb -i [OPTIONS]`

8.3.1.1 Customer Managed ORDS Command Options

Option	Description
<code>admin-user <DATABASE USER></code>	The administrator database user with privileges to create users and grant privileges to database users in the Autonomous Database.
<code>db-pool <POOL NAME></code>	The name of the database connection pool.
<code>db-user <DATABASE USER></code>	The ORDS runtime database user.
<code>gateway-user <DATABASE USER></code>	The PLSQL gateway database user that has privileges to access the stored procedures.
<code>feature-db-api <BOOLEAN></code>	Specifies if you want to enable DB API feature. Possible values are <code>true</code> or <code>false</code> . <ul style="list-style-type: none"> • If the value is set to <code>true</code>, then DB API feature is enabled. • If the value is set to <code>false</code>, then DB API feature is disabled. Returns an error if the specified options are <code>--feature-sdw true</code> and <code>--feature db-api false</code> .
<code>feature-rest-enabled-sql <BOOLEAN></code>	Specifies if you want to enable REST-Enabled SQL feature. Possible values are <code>true</code> or <code>false</code> . <ul style="list-style-type: none"> • If the value is set to <code>true</code>, then the REST-Enabled SQL feature is enabled. • If the value is set to <code>false</code>, then the REST-Enabled SQL feature is disabled. Returns an error if the specified options are <code>--feature-sdw true</code> and <code>--feature-rest-enabled-sql false</code> .
<code>feature-sdw <BOOLEAN></code>	Specifies if you want to enable Database Actions feature. Possible values are <code>true</code> or <code>false</code> . <ul style="list-style-type: none"> • If the value is set to <code>true</code>, then the Database Actions feature is enabled. • If the value is set to <code>false</code>, then the Database Actions feature is disabled. If the option is set to <code>true</code> , then the following settings are set to <code>true</code> in the configuration file: <ul style="list-style-type: none"> • <code>database.api.enabled</code> • <code>restEnabledSql.active</code> Returns an error if <code>--feature-sdw true</code> and any of following options are specified, and are set to <code>false</code> : <ul style="list-style-type: none"> • <code>--feature-db-api</code> • <code>--feature-rest-enabled-sql</code>

Option	Description
<code>-h, --help</code>	Shows usage information for a command.
<code>-i, --interactive</code>	Prompts for the required information.
<code>log-folder <FOLDER></code>	Writes the logs in the folder when creating the users and granting privileges to the user. If this option is omitted, then the output is written to standard output.
<code>password-stdin</code>	Reads the password value from standard input when redirecting input to a file or here document.
<code>wallet <PATH></code>	The location of the wallet zip file downloaded from Autonomous Database. Returns an error if the wallet is omitted and the <code>db.wallet.zip.path</code> setting does not exist in the ORDS configuration.
<code>wallet-service-name <NET SERVICE NAME></code>	The net service name in the <code>tnsnames.ora</code> file located in the wallet zip file. If <code>--wallet-service-name</code> option is omitted and the setting <code>db.wallet.zip.service</code> does not exist in ORDS configuration, then it defaults to <code><db>_LOW</code> that is got from the <code>tnsnames.ora</code> file.

8.3.1.2 Interactive Installation Prompts

This section describes the interactive installation prompts to setup your Customer Managed ORDS.

To setup your Customer Managed ORDS, use the ORDS Command-Line Interface (CLI) to interactively prompt you for the information.

Example:

```
ords install adb --interactive --prompt-password --log-folder <LOG
  FOLDER>
```

Where:

- `--prompt-password`: prompt you for the runtime database user's password and the gateway database user's passwords
- `--prompt-password`: If this option is omitted, then the passwords are generated. Special care should be considered for database user's password. If you plan to use ORDS on multiple servers and use the same runtime database user and gateway database user, then specify the `--prompt-password` option to ensure that the same passwords are being used.

Table 8-1 Interactive Installation Prompts

Pro mpt Num ber	Prompt	Description
1.	Enter a number to select the database pool to update, or create an additional database pool. The selected (or created) database pool will be used to configure a Customer Managed ORDS. [1] default MYADB_MEDIUM /path/to/myadb/wallet.zip [2] Create an additional database pool Choose [1]:	Refer to Entering a Number to Select the Database Pool
2.	Enter the database pool name:	Refer to Entering the Database Pool Name
3.	Enter the Autonomous Database Wallet path: /path/to/wallet.zip	Refer to Entering the Autonomous Database Wallet Path
4.	Enter a number to select the TNS Network alias to use [1] DEMO_LOW ...service_name=g123_demo_low.adb.oraclecloud... [2] DEMO_MEDIUM ...service_name=g123_demo_medium.adb.oracleclo... [3] DEMO_HIGH ...service_name=g123_demo_high.adb.oraclecloud... Choose [1]: 2	Refer to Enter a Number to Select the TNS Network Alias
5.	Provide database user name with administrator privileges. Enter the administrator username [ADMIN]:	Enter the Administrator Username
6.	Enter the database password for ADMIN:	Enter the Database Password for ADMIN
7.	Enter the ORDS runtime database username [ORDS_PUBLIC_USER2]:	Entering the ORDS Runtime Database Username
8.	Enter the database password for ORDS_PUBLIC_USER2: Confirm password:	Entering the Database Password for ORDS_PUBLIC_USER2

Table 8-1 (Cont.) Interactive Installation Prompts

Pro mpt Num ber	Prompt	Description
9.	Enter the PL/SQL Gateway database username:	Entering the PL/SQL Gateway Database Username
10.	Enter the database password for ORDS_PLSQL_GATEWAY2: Confirm password:	Enter the Database Password for ORDS_PLSQL_GATEWAY2
11.	Connecting to Autonomous database user: ADMIN TNS Service: DEMO_MEDIUM Retrieving information	
12.	Enter a number to select additional feature(s) to enable: [1] Database Actions (Enables all features) [2] REST Enabled SQL and Database API [3] REST Enabled SQL [4] Database API [5] None Choose [1]:	Entering a Number to Select and Enable Additional Feature
13.	Enter a number to configure and start ORDS in standalone mode [1] Configure and start ORDS in standalone mode [2] Skip Choose [1]:	Enter a Number to Configure and Start ORDS
14.	Enter a number to select the protocol [1] HTTP [2] HTTPS Choose [1]: 2	Entering a Number to Select a Protocol
15.	Enter the HTTP port [8080]:	Entering the HTTP Port
16.	Enter the HTTPS port [8443]:	Entering the HTTP Port

Table 8-1 (Cont.) Interactive Installation Prompts

Pro mpt Num ber	Prompt	Description
17.	Enter the APEX static resources location: /path/to/apex/images	Entering the APEX Static Resources Location

8.3.1.2.1 Entering a Number to Select the Database Pool

This prompt is displayed, if an ORDS configuration exists and contains database pool(s).

You can select a database pool to update, or create an additional database pool for your Customer Managed ORDS.

If option 2 is selected, then Prompt number 2 is displayed. Otherwise, Prompt number 3 is displayed.



Note:

If this is the first time you are setting up the Customer Managed ORDS, and the ORDS configuration does not exist, then you are prompted for the wallet location. See Prompt number 3.

8.3.1.2.2 Entering the Database Pool Name

Specify the database pool name.

8.3.1.2.3 Entering the Autonomous Database Wallet Path

Specify the location and filename of the downloaded Autonomous Database wallet.



Note:

If this is the first time you are setting up the Customer Managed ORDS, then you are prompted for the wallet location. Otherwise, if an ORDS configuration already exists, then Prompt number 1 is displayed.

8.3.1.2.4 Enter a Number to Select the TNS Network Alias

Select the TNS alias name that was retrieved from the `tnsnames.ora` file contained in the wallet zip file.

8.3.1.2.5 Enter the Administrator Username

Specify a database user with administrator privileges. Defaults the database user to ADMIN.

8.3.1.2.6 Enter the Database Password for ADMIN

Specify the password for administrator database user.

8.3.1.2.7 Entering the ORDS Runtime Database Username

Specify the ORDS runtime database user. Defaults the database user to `ORDS_PUBLIC_USER2`.

8.3.1.2.8 Entering the Database Password for ORDS_PUBLIC_USER2

Prompts for the password if `--prompt-password` option is specified on the command line. Otherwise, the password prompt is not displayed, and the password is generated.

 **Note:**

If the runtime database user does not exist in the Autonomous Database, then the runtime database user is created and granted privileges. If the runtime database user already exists in the Autonomous Database, and the runtime user's password does not match the password in the Autonomous Database, then the runtime database user password is changed.

8.3.1.2.9 Entering the PL/SQL Gateway Database Username

Specify the ORDS PL/SQL gateway database user.

8.3.1.2.10 Enter the Database Password for ORDS_PLSQL_GATEWAY2

Prompts for the password if `--prompt-password` option is specified on the command line. Otherwise, the password prompt is not displayed, and the password is generated.

 **Note:**

If the PL/SQL gateway database user does not exist in the Autonomous Database, then the gateway database user is created and granted privileges. If the gateway user already exists in the Autonomous Database, and the gateway user's password does not match the password in the Autonomous Database, then the gateway database user password is changed..

8.3.1.2.11 Entering a Number to Select and Enable Additional Feature

Select the additional feature that you want to enable.

 **See Also:**

- About Oracle Database Actions
- REST- Enabled SQL Service
- ORDS Database API

8.3.1.2.12 Enter a Number to Configure and Start ORDS

You can configure ORDS to run in standalone mode. In addition, you can start ORDS in standalone mode after setup is completed for Customer Managed ORDS.

8.3.1.2.13 Entering a Number to Select a Protocol

Select a protocol.

- If option 1 is selected, then prompt number 15 is displayed.
- If option 2 is selected, then prompt number 16 is displayed.

8.3.1.2.14 Entering the HTTP Port

Specify the HTTP port.

8.3.1.2.15 Entering the HTTPS Port.

Specify the HTTPS port.

8.3.1.2.16 Entering the APEX Static Resources Location

This prompt displays only if you are maintaining the APEX static resources.

Note:

If you are using the Oracle Content Delivery Network for the APEX static resources, then this prompt is not displayed.

See Also:

[Oracle APEX Static Resources](#)

8.3.2 Silent Installation of ADB on Command-Line Interface

For silent installation, provide the following command options to setup your Customer Managed ORDS:

- Database Pool: If this option is omitted, then the `default` database pool is used.
- Wallet Path: This is required if this option does not exist in the ORDS configuration database pool.
- Wallet Service Name: The TNS alias name from `tnsnames.ora` file contained in the wallet zip file. If this option is omitted, and the setting `db.wallet.zip.service` does not exist in the ORDS configuration database pool, then the wallet service name defaults to `<DB>_LOW`.
- Administrator username and password (Required)
- Runtime database username and password (Required)
- PL/SQL gateway username and password (Required)
- Additional Database Features

Install ADB Command

```
ords install adb [OPTIONS]
```



See Also:

[Customer Managed ORDS Command Options](#)

8.3.2.1 Using Input Redirection

This section describes how to redirect the standard input using the Here document or to a file.

Redirect STDIN to a file

Redirect STDIN to a file that contains the password. In the following example, the file must contain three passwords. Each password must be on a separate line.

Example:

```
ords install adb --admin-user <DATABASE USER> --db-user <DATABASE USER> --  
gateway-user <DATABASE  
USER> --wallet <PATH> --wallet-service-name <NET SERVICE NAME> --  
feature-sdw <BOOLEAN> --log-folder  
<FOLDER> --password-stdin < filename.txt
```

Where the `filename.txt` contains passwords:

```
<PASSWORD FOR admin-user>  
<PASSWORD FOR db-user>  
<PASSWORD FOR gateway-user>
```

Starting from left to right, the first password belongs to the first user option (`--admin-user`) on the command line. The second password belongs to the second user option on the command line (`--db-user`) and the third password belongs to the third user (`--gateway-user`) option on the command-line.

Redirect Standard Input Using Here Document

Redirect STDIN using the Here document (also known as heredoc) for the password(s). The heredoc consists of the `<<<` redirection operator followed by a delimiter token.

Each password must be on a separate line and it is ended by the delimiter token.

Example:

```
ords install adb --admin-user <DATABASE USER> --db-user <DATABASE USER> --  
gateway-user <DATABASE USER>  
--wallet <PATH> --wallet-service-name <NET SERVICE NAME> --feature-sdw  
<BOOLEAN> --log-folder <FOLDER>  
--password-stdin << EOF  
<PASSWORD FOR admin-user>  
<PASSWORD FOR db-user>
```

```
<PASSWORD FOR gateway-user>  
EOF
```

Starting from left to right, the first password belongs to the first user option (`--admin-user`) on the command line. The second password belongs to the second user option on the command line (`--db-user`) and the third password belongs to the third user (`--gateway-user`) option on the command-line. The Here document is ended by the token EOF.



Note:

Once the operation is complete, delete the file or script that contains the passwords.

9

Oracle API for MongoDB Support

This section describes the support for the Oracle Database API for MongoDB.

Starting with ORDS release 22.3, Oracle REST Data Services supports the Oracle Database API for MongoDB when running in a standalone mode. This enables the use of MongoDB drivers, frameworks, and tools to develop your JSON document-store applications against the Oracle Database. The Oracle Database API for MongoDB, translates the MongoDB wire protocol into SQL statements that are executed using the ORDS connection pools.

Figure 9-1 Architecture Diagram for Oracle Database API for MongoDB



Following points must be considered:

- The MongoDB concept of a database is mapped to the concept of a schema in Oracle Database. Specifically, an ORDS-enabled schema. For example, when you insert a JSON document into a collection in the database `foo`, the API for MongoDB inserts the document into a collection in the ORDS-enabled schema `foo`.
- Authentication and authorization when using the API for MongoDB depends on the Oracle Database users and access controls and not the MongoDB users. When you are connecting a MongoDB client, you must specify the Oracle Database credentials using the LDAP authentication mechanism of MongoDB using the connection options `authMechanism=PLAIN` and `authSource=$external`. The protocol passes the username and password as a plain text and so, TLS/SSL must be enabled using the connection option `tls=true`.
- The Oracle API for MongoDB maps the MongoDB commands to the corresponding SQL statements on the table backing the collection. For example, a MongoDB command such as `emp.find({"name":"John"})` is executed against the database using a SQL select statement similar to `select data from emp e where e.data."name" = 'John'`.
- When required, SQL can be executed directly over JSON collections. With this model, you get the speed, flexibility, and ease-of-use of a NoSQL document store while still having the ability to use SQL for analytics and reporting directly over your natively stored JSON data.

 **See Also:**

- [Oracle Database API for MongoDB](#)
- [Interactive Command-Line Interface Installation](#)
- [Serve Commands for Running in Standalone Mode](#)

9.1 Getting Started

Perform the following steps to get started to use the Oracle Database API for MongoDB:

1. Install and configure ORDS:

From a command prompt, install and configure ORDS using the following commands:

 **Note:**

If you are installing ORDS against the Autonomous Database, then use the command `ords install adb`.

 **See Also:**

[Installing and Configuring Customer Managed ORDS on Autonomous Database](#)

```
ords install
ords config set mongo.enabled true
ords serve
```

Log showing that the Oracle API for MongoDB is enabled:

After starting ORDS using the `ords serve` command, the log shows a message similar to the following to verify if the Oracle API for MongoDB is enabled:

```
Disabling document root because the specified folder does not exist: ./
config/global/doc_root
2022-08-17T15:23:04.043Z INFO          Oracle API for MongoDB listening on
port: 27017
2022-08-17T15:23:04.050Z INFO          The Oracle API for MongoDB connection
string is:
mongodb://[{}user]:{password}@localhost:27017/{user}?
authMechanism=PLAIN&authSource=$external&ssl=true&retryWrites=false&loadBal
anced=true
```

2. Create an ORDS-enabled user:

Leave the server running and from SQLcl, create an ORDS-enabled user:

```
create user foo identified by "MyPassword1!";
grant soda_app, create session, create table, create view, create
sequence, createprocedure, create job, unlimited tablespace to foo;
```



```
connect foo/MyPassword1!
exec ords.enable_schema;
```

3. Connect to ORDS using the MongoDB shell:

Note:

By default, ORDS uses a self-signed certificate and so, `tlsAllowInvalidCertificates` is necessary. A signed certificate can be configured by setting the properties `standalone.https.cert` and `standalone.https.cert.key`.

```
mongosh --tlsAllowInvalidCertificates 'mongodb://foo:MyPassword1!
@localhost:27017/foo?
authMechanism=PLAIN&authSource=$external&tls=true&retryWrites=false&loadBal
anced=true'
foo> db.createCollection('emp');
{ ok: 1 }
foo> db.emp.insertOne({"name":"Blake","job": "Intern","salary":30000});
...
foo> db.emp.insertOne({"name":"Smith","job": "Programmer","salary":
60000,"email" : "smith@oracle.com"});
...
foo> db.emp.insertOne({"name":"Miller","job": "Programmer","salary":
70000});
...
foo> db.emp.find({"name":"Miller"});
[
  {
    _id: ObjectId("6320bfc40dd73b60ef5641b9"),
    name: 'Miller',
    job: 'Programmer',
    salary: 70000
  }
]

foo> db.emp.updateOne({"name":"Miller"}, {$set:
{"email":"miller@oracle.com"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

4. The data inserted from the MongoDB client can be accessed from SQL:

```
SQL> select json_serialize(e.data)
       from emp e;
```

```
JSON_SERIALIZE(E.DATA)
```

```

-----
-----
{"_id":"6320bfa30dd73b60ef5641b7","name":"Blake","job":"Intern","salary":30
000}
{"_id":"6320bfb30dd73b60ef5641b8","name":"Smith","job":"Programmer","salary
":60000,"email":"smith@oracle.com"}
{"_id":"6320bfc40dd73b60ef5641b9","name":"Miller","job":"Programmer","salar
y":70000,"email":"miller@oracle.com"}

SQL> select e.data."name".string() n,
          e.data."job".string() j
        from emp e
        where e.data."email".string() = 'miller@oracle.com';

N                J
-----
Miller           Programmer

```

9.2 Requirements

This section lists the client and database requirements.

The MongoDB API supports Oracle Database version 21c or later and the Autonomous Oracle Database 19c or later (serverless, dedicated, and cloud@customer). In general, Oracle API for MongoDB supports MongoDB tools and drivers that support the `loadBalanced` connection option. Oracle API for MongoDB specifically supports the client versions listed in the following table.

Table 9-1 Requirements

Client	Minimum Version Supported
mongosh	0.15.6
Java	4.3.0
Python	3.12.0
Node.js	4.1.0
C#	2.13.0
Golang	1.6.0
Database Tools(mongoimport, mongorestore)	100.5.2

9.3 Configurable Settings for MongoDB

This section lists the editable configuration settings to support the MongoDB API stored in the global configuration, located at `global/settings.xml`.

Note:

Oracle recommends users to use the Oracle REST Data Services command-line interface to edit the configuration files.

Table 9-2 Configuration Settings to Support the Mongo API

Key	Type	Description
mongo.enabled (Mandatory property)	boolean	Specifies to enable the API for MongoDB. Default value is <code>false</code> . To enable the API for MongoDB, set the value to <code>true</code> .
mongo.access.log	path	Specifies the path to the folder where you want to store the API for MongoDB access logs. If the path is not specified, then no access is generated.
mongo.host	string	Specifies a comma separated list of host names or IP addresses to identify a specific network interface on which to listen. Default value is <code>0.0.0.0</code> .
mongo.port	integer	Specifies the API for MongoDB listen port. Default value is <code>27017</code> .
mongo.idle.timeout	duration	Specifies the maximum idle time for a connection in milliseconds. Default value is <code>30m</code>
mongo.op.timeout	duration	Specifies the maximum time for a database operation in milliseconds. Default value is <code>10m</code> .

**See Also:**

- [About the Oracle REST Data Services Configuration Files](#)

9.4 Examples

This section lists some examples that use the `ords config set` command to store the MongoDB settings in the current working directory (CWD) `global/settings.xml` and also provides an example for MongoDB listener settings in `global/settings.xml` file.

Examples of using the `ords config set` command

- `ords config set mongo.enabled true`
- `ords config set mongo.host example.com`
- `ords config set mongo.port 27017`
- `ords config set mongo.idle.timeout 40m`
- `ords config set mongo.op.timeout 15m`

Example of Mongo Listener settings in `global/settings.xml`

```
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>Saved on Wed Jun 15 01:33:58 UTC 2022</comment>
<entry key="debug.printDebugToScreen">true</entry>
. . .
<entry key="mongo.host">example.com</entry>
<entry key="mongo.idle.timeout">40m</entry>
<entry key="mongo.enabled">true</entry>
```

```
<entry key="mongo.op.timeout">15m</entry>
<entry key="mongo.port">27016</entry>
. . .
</properties>
```

Examples of using the ords config get command

- `ords config get mongo.enabled`
true
- `ords config get mongo.port`
27016

Using the ords serve Command

Use the `ords serve` command to run in standalone mode.

Note:

To enable the Oracle API for MongoDB, prior to running the `serve` command, you must set `mongo.enabled` property to `true`

See Also:

- [Setting Up the Configuration Folder Structure](#)
- [Serve Commands for Running in Standalone Mode](#)

9.5 Accessing the Connection Pools

Oracle REST Data Services supports the ability to connect to more than one database. Multiple named pools can be defined using the `install` command. Adding a pool creates a corresponding directory under `./databases` within the ORDS configuration directory. An initial install of Oracle REST Data Services typically adds a default database connection pool named `default`.

By default, when you are connecting to the MongoDB API, all database requests are directed to the default connection pool. Optionally, you can route MongoDB API requests to other connection pools using the `hostnames` file. For example, assume that the database pool named `mydb1` has a configuration directory at `ords_config/databases/mydb1`. If you create the `hostname` file `ords_config/databases/mydb1/hostnames` containing two hostnames as follows:

```
www.example.com
example.com
```

Then, the following connection strings routes to `mydb1` connection pool instead of the default connection pool:

```
"mongodb://www.example.com:27017/scott?  
authMechanism=PLAIN&authSource=$external&ssl=true&retryWrites=false&loadBalanc  
ed=true"
```

```
"mongodb://example.com:27017/scott?  
authMechanism=PLAIN&authSource=$external&ssl=true&retryWrites=false&loadBalanc  
ed=true"
```



See Also:

- [Routing Based on the Request Hostname](#)

9.6 Logging MongoDB API Access

This section describes how to enable logging of requests to the MongoDB API.

By default, requests to the MongoDB API are not logged. To enable logging access to MongoDB API, you must set the configuration property `mongo.access.log` to a directory path. If the directory path is not absolute, it is resolved relative to the the ORDS configuration directory (`<ords config>`). If the directory does not exist, then ORDS creates the directory on startup. ORDS then adds an access log file entry within this directory each time the MongoDB API is accessed.

Example:

```
ords config set mongo.access.log mongologs
```

This command writes access log files under `<ords config>/mongologs/` folder.

9.7 Achieving High Performance

This section describes the settings that are useful in achieving high performance.

In environments where higher performance or throughput is desired, some ORDS connection pool parameters are required to be configured and tuned.

To achieve higher throughput, the following settings and the values are useful. The optimal tuning of these parameters depend on the requirements of an application:

```
ords config set jdbc.MaxConnectionReuseCount 5000  
ords config set jdbc.MaxConnectionReuseTime 900  
ords config set jdbc.SecondsToTrustIdleConnection 1  
ords config set jdbc.InitialLimit 100  
ords config set jdbc.MaxLimit 100
```

Where:

- `MaxConnectionReuseTime`: Enables connections to be gracefully closed and removed from the connection pool after a connection is borrowed a specific number of times.

- `SecondsToTrustIdleConnection`: Sets the time in seconds to trust a recently used or recently tested database connection and skip the validation test during connection checkout.
- `InitialLimit` and `MaxLimits`: Sets the connection pool size in the UCP for the specified connection pool.

10

Migration of Legacy Configuration

This section describes how to migrate the legacy configuration and the URL mapping.

Legacy configuration is any configuration of ORDS for versions starting with 2.x to version 21.4.x. These configurations need to be migrated to the new directory structure for 22.1 and higher versions of ORDS.

10.1 Migrating From Legacy Configurations Using the CLI Installation

To migrate your applications from legacy configurations, you can use the `install` command or the `migrate config` command.

You can migrate your applications from legacy configurations in the following two ways:

Migrating the Configuration and Upgrading ORDS

You use this option when you migrate your legacy configurations (ORDS 21.4.x or earlier versions) to a new configuration structure, and also upgrade ORDS in the database. You can perform this using one of the following two options:

Using the Interactive Install Command

If you use the interactive `install` command, then it migrates the legacy configurations to the new configuration structure. Then, depending on the options that you selected in the earlier steps of installation, it prompts you to either upgrade ORDS in the database, or create or update additional database pools.

```
ords --config <configuration_folder> install --interactive  
--legacy-config <legacy_config_folder> --log-folder <log_folder>
```

Using the Non-Interactive Install Command

If you use the non-interactive `install` command, then you must specify the required information to migrate the configuration in the command line.

```
ords --config <configuration_folder> install [--db-pool <pool_name>] --admin-  
user <user>  
--legacy-config <folder> --log-folder <log_folder> --password-stdin < <text-  
file>
```

Migrating the Configuration Only

You use this option when you only migrate your legacy configurations (ORDS 21.4.x or earlier version) to a new configuration structure, but do not upgrade ORDS to a newer version.

```
ords --config <configuration_folder> migrate config <legacy_config_folder>
```

When migrating the configuration completes, the following message is displayed:

```
Legacy configuration files located at <legacy_config> are no longer being
used to configure ORDS
  or its connection pools. Your migrated configuration files are now located
at <configuration_folder>.
```

10.2 Migrating URL Mapping

The legacy URL mapping file is `/path/to/conf/url-mappings.xml` that contains different types of mappings.

The installer migrates the url-mappings. For each url-mapping that contains the following type, it performs the following:

- `base-path` type: creates a path file for that database-pool containing the base path (single or multiple path segments)
- `base-url` type: creates a hostname file for that database-pool containing the hostname.

The base-path mapping pattern may contain a path that consists of a single segment or multiple segments.

Example:

- `/sales`
- `/pls/hr`

The base-url mapping pattern may contain only the hostname, or it can include the protocol, hostname, and path.

Example:

```
https://example.com/ords/support
```

```
www.sales.com
```

The ORDS install CLI migrates the `url-mapping.xml` as follows:

For each base-path type mapping (for example, `/sales`), the path is added to the paths file of the pool.



Note:

If the paths file does not exist, it is created.

For each base-url type mapping (for example `https://example.com/ords/support`), the hostname (for example `example.com`) is added to the hostnames file of the pool.

If the base-url mapping contains a path after the context (for example, `/ords`), then the path (for example, `/support`) is added to the paths file.



Note:

If the hostnames or paths files do not exist, then it is created.

 **Note:**

A SEVERE warning message is displayed if:

- Duplicate paths or hostnames across different database pools exist
- The database pool that the mapping points to does not exist
- The hostname or path is invalid

 **Note:**

The name of the directory that contains the `pool.xml` configuration file is used in the request URL path mapping. When you are migrating a `url-mapping.xml` file, the pool name from that file is used as the directory name. There is a possibility that a pool name overlaps with an entry in the paths file from another pool definition. The pool name takes precedence in the request URL mapping.

The following is an example `url-mapping.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<pool-config xmlns="http://xmlns.oracle.com/apex/pool-config">
  <pool name="db1" base-path="/sales" updated="2021-09-23T19:22:19.944Z"/>
  <pool name="db1" base-path="finance/" updated="2021-09-23T19:22:50.189Z"/>
  <pool name="db2" base-url="https://example.com/ords/support"
updated="2021-09-23T19:23:04.751Z"/>
  <pool name="db3" base-path="/pls/hr" updated="2021-09-23T19:23:14.832Z"/>
  <pool name="db4" base-path="employee" updated="2021-09-23T19:23:24.252Z"/>
  <pool name="db5" base-url="www.sales.com"
updated="2021-09-23T19:23:35.322Z"/>
</pool-config>
```

Example after migrating the url-mappings to the database pools folder:

```
+-- databases/
  +- db1/
    +- pool.xml
    +- paths // contains: [sales, finance]
  +- db2/
    +- pool.xml
    +- hostnames // contains: [example.com]
    +- paths // contains: [support]
  +- db3/
    +- pool.xml
    +- paths // contains: [/pls/hr]
  +- db4/
    +- pool.xml
    +- paths // contains: [employee]
  +- db5/
    +- pool.xml
    +- hostnames // contains: [www.sales.com]
```

A

Non-Interactive ORDS CLI Commands Examples

This section provides some examples for non-interactive install/upgrade, and uninstall.

The following are examples for non-interactive install/upgrade, and uninstall.

The password file must contain the password(s) in the correct order to match the order of the user(s), option(s) that are specified on the command-line.

The following examples can be used for a PDB, CDB. Ensure that the `--db-servicename` option contains the correct value for the servicename for the CDB, PDB.

If the `--log-folder <FOLDER>` option is specified and the log folder does not exist, then the folder is created.

A.1 Non-Interactive Install/Upgrade

This section explains how to create a proxied APEX_PUBLIC_USER.

If APEX is installed and the APEX_PUBLIC_USER exists in the database, then the APEX_PUBLIC_USER is automatically configured to be a proxied user.

```
cat password.txt  
  
<password for admin-user>  
  
<password for proxy-user>  
  
ords --config /path/to/test/config install --db-pool pdb1 --admin-user SYS --  
proxy-user --db-hostname localhost --db-port 1521 --db-servicename orcl --  
feature-sdw true --log-folder /path/to/logs --password-stdin < password.txt
```

A.2 Non-Interactive Install/Upgrade (create a proxied PLSQL Gateway User)

This section explains how to create a proxied PLSQL Gateway User.

Specify a PL/SQL gateway user and `--gateway-mode proxied`.

```
ords --config /path/to/test/config install --db-pool pdb2 --admin-user SYS --  
proxy-user --db-hostname localhost --db-port 1521 --db-servicename pdb2 --  
feature-sdw true --gateway-user EXAMPLE_USER1 --gateway-mode proxied --log-  
folder /path/to/logs --password-stdin < password.txt
```

A.3 Non-Interactive Install/Upgrade (specify PLSQL Gateway User and not using RESTful Services)

This section explains how to specify PLSQL Gateway User not using RESTful Services.

Specify a PL/SQL gateway user and `--gateway-mode direct`. The direct mode indicates that you are only using the PL/SQL GATEWAY and not RESTful Services. The gateway user password is required for direct mode.

```
cat gatewaypwd.txt

<password for admin-user>

<password for proxy-user>

<password for gateway-user>

ords --config /path/to/test/config install --db-pool pdb3 --admin-user SYS --
proxy-user --db-hostname localhost --db-port 1521 --db-servicename pdb3 --
gateway-user EXAMPLE_USER2 --gateway-mode direct --log-folder /path/to/logs --
password-stdin < gatewaypwd.txt
```

A.4 Non-Interactive Install/Upgrade (disable a configured PLSQL Gateway)

This section explains how to disable a configured PLSQL Gateway.

Disables the proxied PL/SQL gateway user.

```
ords --config /path/to/test/config install --db-pool pdb2 --admin-user SYS --
proxy-user --db-hostname localhost --db-port 1521 --db-servicename pdb2 --
feature-sdw true --gateway-mode disabled --log-folder /path/to/logs --password-
stdin < password.txt
```

A.5 Non-Interactive Uninstall ORDS

This section explains how to uninstall ORDS in the database.

To uninstall ORDS, execute the following commands:

```
cat adminpwd.txt
<password for admin-user>

ords --config /path/to/test/config uninstall --admin-user SYS --db-hostname
localhost --db-port 1521
--db-servicename orcl --log-folder /path/to/logs < adminpwd.txt
```

A.6 config CLI Examples

This section provides config CLI examples.

Non-Interactive Update Wallet

If you want to change the password in the wallet, then specify the following:

```
ords --config /path/to/test/config config secret --password-stdin db.password <
userpwd.txt
```

If you want to view the password in the wallet, then specify the following:

```
ords --config /path/to/test/config config get --secret db.password
```

Where:

`userpwd.txt` contains the password.



Note:

If you are changing the wallet for a specific pool, then you must provide the `--db-pool` option. The preceding example updates the default pool.

B

ORDS Legacy Commands and New ORDS CLI Commands

Table B-1 ORDS Legacy and New CLI Commands

Legacy Command	New CLI Command	Description
<code>java -jar ords.war configdir</code>	<code>ords --config /path/to/config <command></code> Example of using config option: <code>ords --config /path/to/config install <OPTIONS></code> <code>ords --config /path/to/config uninstall <OPTIONS></code>	<p>Specifies the ORDS configuration folder that contains configuration files.</p> <p>New CLI Command: If the configuration folder is not specified, your current working directory is your configuration folder. The command does not store the configuration location. You must provide the configuration location or use the default configuration location.</p> <p>Legacy configuration directory: The configuration location is stored in <code>ords.war</code> file. When the user executes the <code>ords</code> command, (example: <code>install</code>, <code>setup</code>) and the configuration location exists in <code>ords.war</code> file, then it does not prompt for the configuration location.</p> <p>Refer to Setting Up the Configuration Folder Structure for more information.</p>

Table B-1 (Cont.) ORDS Legacy and New CLI Commands

Legacy Command	New CLI Command	Description
<pre>java -jar ords.war install advanced</pre>	<pre>/ords install ords --config /path/to/config install ords --config /path/to/config install --interactive ords --config /path/to/config install -i</pre>	<p>Install ORDS in interactive mode.</p> <p>Legacy install advanced: Prompts for the configuration information only if the ORDS configuration does not exist.</p> <p>Prompts if you want to run in a standalone mode.</p>
<pre>java -jar ords.war install --silent java -jar ords.war install simple java -jar ords.war</pre>	<pre>ords --config /path/to/config install <OPTIONS></pre>	<p>Install ORDS in a non-interactive mode. No prompting for new install CLI command if the user provides install options and not specifying the <code>-i</code> or <code>--interactive</code> option.</p> <p>Legacy install simple or no command : (<code>java -jar ords.war</code>)</p> <p>Prompts for the configuration information only if the ORDS configuration does not exist.</p> <p>If the configuration file does not exist and if the default parameter file does not contain any configuration values, then it prompts.</p>

Table B-1 (Cont.) ORDS Legacy and New CLI Commands

Legacy Command	New CLI Command	Description
java -jar ords.war schema	ords --config /path/to/config install	Install or upgrade ORDS only. The configuration file is not updated. New CLI Command: Enter a number to select the type of installation [1] Install or upgrade ORDS in the database only [2] Create or update a database pool and install/upgrade ORDS in the database [3] Create or update a database pool only Choose [1]: Option 1 is equivalent.
java -jar ords.war uninstall advanced	ords uninstall ords --config /path/to/config uninstall	Uninstalls ORDS in interactive mode.
java -jar ords.war uninstall	ords --config /path/to/config uninstall --interactive ords --config /path/to/config uninstall -i	
java -jar ords.war uninstall --silent	ords --config /path/to/config uninstall <OPTIONS>	Uninstall ORDS in non-interactive mode. If you provide uninstall options and not specify the -i or --interactive option, then you are not prompted for new uninstall CLI command.
java -jar ords.war uninstall simple		
java -jar ords.war set-property <KEY> <VALUE>	ords --config /path/to/config config set <KEY> <VALUE>	Updates the setting in the configuration file. See Updating the Configuration Settings for a list of options.
java -jar ords.war standalone	ords --config /path/to/config serve	Launch ORDS in standalone mode.
java -jar ords.war user <USERNAME> ROLES	ords --config /path/to/config config user add <NAME> <ROLE>	Create the user and its roles to provide permission to the ORDS REST Services.
java -jar ords.war validate	ords install repair ords --config /path/to/conf install repair ords --config /path/to/conf install repair --interactive ords --config /path/to/conf install repair -i	Repair ORDS in interactive mode.

Table B-1 (Cont.) ORDS Legacy and New CLI Commands

Legacy Command	New CLI Command	Description
<code>java -jar ords.war validate --silent</code>	<code>ords --config /path/to/conf install repair <OPTIONS></code>	Repair ORDS in non-interactive mode. There is no prompting for install repair CLI command if user provides install options after omitting the <code>-i</code> or <code>--interactive</code> option.

C

About the Oracle REST Data Services Configuration Files

The section describes the Oracle REST Data Services configuration files.

Topics:

- [Understanding the Configuration Folder Structure](#)
- [Understanding the Configuration File Format](#)
- [Understanding the Configurable Settings](#)

C.1 Understanding the Configuration Folder Structure

The configuration folder has the following structure:

```
+-- global/
  +- settings.xml
  +- credentials
  +- wallet/
  +- standalone/
+- databases/
  +- default/
    +- pool.xml
    +- wallet/
  +- myapp/
    +- hostnames
    +- pool.xml
    +- wallet/
  +- myapp2/
    +- paths
    +- pool.xml
    +- wallet/
```

The `global/` folder contains settings that apply across the entire ORDS instance:

- `settings.xml`: Contains settings that are configured across the entire ORDS instance. For **example**: `debug.printToScreen=true`
- `credentials`: The ORDS user password file
- `wallet/`: Contains an Oracle auto login wallet that contains the instance wide encryption and mac keys previously stored in `security.crypto.enc.password` and `security.crypto.mac.password` configuration settings in `defaults.xml`.
- `standalone/`: Contains standalone mode related resources such as the HTTPS certificate and key.

The `databases/` folder contains database pool configurations.

- Each pool configuration is located in its own folder. The base path url mapping for a pool is inferred from the folder name. If the folder is named `foo/`, then requests can be mapped to the pool by accessing `https://server/ords/foo/...`
- The database pool folder name must only contain lowercase alphabet a-z, digits 0-9, '-', '.' or '_' character.
- The folder named `default/` is reserved and is used to map requests that are not mapped to any other pool. It is equivalent to the `apex.xml` pool in the old structure.
- The folder named `databases/<pool-name>/wallet/` contains an Oracle auto login wallet that contains the credentials for the database pool. The database username and password must be stored in the wallet. The `db.password` settings must not be used. The wallet must conform to the requirements for ORDS wallets.
- Alternatively, the folder may contain a file named `hostname` or `paths`, but not both.

C.2 Understanding the Configuration File Format

Configuration files use the standard Java XML properties file format, where each configuration setting contains a key and a corresponding value. The following is an example of `settings.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<entry key="db.connectionType">basic</entry>
<entry key="db.hostname">localhost</entry>
<entry key="db.port">1521</entry>
<entry key="db.servicename">orcl</entry>
<entry key="jdbc.DriverType">thin</entry>
<entry key="jdbc.InitialLimit">3</entry>
<entry key="jdbc.MinLimit">1</entry>
<entry key="jdbc.MaxLimit">10</entry>
<entry key="jdbc.MaxStatementsLimit">10</entry>
<entry key="jdbc.InactivityTimeout">1800</entry>
<entry key="jdbc.statementTimeout">900</entry>
<entry key="jdbc.MaxConnectionReuseCount">1000</entry></properties>
```

C.3 Understanding the Configurable Settings

This section lists the editable configuration settings for the `settings.xml` and `pool.xml` configuration files.



Note:

Oracle recommends users to use the Oracle REST Data Services command-line interface to edit the configuration files.

Table C-1 Oracle REST Data Services Configuration Settings

Key	Type	Description	Example	Setting Type
apex.security.administrator.roles	string	Specifies the comma delimited list of additional roles to assign authenticated APEX administrator type users.	Not applicable	Pool specific
apex.security.user.roles	string	Specifies the comma delimited list of additional roles to assign authenticated regular APEX users.	Not applicable	Pool specific
autoupgrade.api.aulocation	string	specifies a configuration setting for AutoUpgrade.jar location.	Not applicable	Pool specific
autoupgrade.api.enabled	boolean	Specifies a configuration setting to enable AutoUpgrade REST API features.	Not applicable	Pool specific
autoupgrade.api.jvmlocation	string	Specifies a configuration setting for AutoUpgrade REST API JVM location.	Not applicable	Pool specific
autoupgrade.api.loglocation	string	Specifies a configuration setting for AutoUpgrade REST API log location.	Not applicable	Pool specific
database.api.management.services.disabled	boolean	Specifies to disable the Database API administration related services. Only applicable when Database API is enabled.	Not applicable	Global
db.adminUser	string	Specifies the username for the database account that ORDS uses for administration operations in the database.	Not applicable	Pool specific
db.adminUser.password	string	Specifies the password for the database account that ORDS uses for administration operations in the database.	Not applicable	Pool specific
db.cdb.adminUser	string	Specifies the username for the database account that ORDS uses for the Pluggable Database Lifecycle Management.	Not applicable	Pool specific
db.cdb.adminUser.password	string	Specifies the password for the database account that ORDS uses for the Pluggable Database Lifecycle Management.	Not applicable	Pool specific

Table C-1 (Cont.) Oracle REST Data Services Configuration Settings

Key	Type	Description	Example	Setting Type
db.credentialSource	string	Specifies the source for database credentials when creating a direct connection for running SQL statements. Value can be one of pool or request. If the value is <code>pool</code> , then the credentials defined in this pool is used to create a JDBC connection. If the value <code>request</code> is used, then the credentials in the request is used to create a JDBC connection and if successful, grants the requestor SQL Developer role. Default value is <code>pool</code> .	Not applicable	Pool specific
cache.metadata.graphql.expireAfterAccess	duration	Specifies the duration after a GraphQL schema is not accessed from the cache that it expires. Default value is 1 minute.	2m	Global
cache.metadata.graphql.expireAfterWrite	duration	Specifies the duration after a GraphQL schema is cached that it expires and has to be loaded again. Default value is 2 minutes.	5m	Not applicable
cache.metadata.jwks.enabled	boolean	Specifies the setting to enable or disable JWKS caching. Supported values: <ul style="list-style-type: none"> <code>true</code> (default) <code>false</code> 	<code>true</code>	Global
cache.metadata.jwks.initialCapacity	numeric	Specifies the initial capacity of the JWKS cache.	10	Global
cache.metadata.jwks.maximumSize	numeric	Specifies the maximum capacity of the JWKS cache.	10000	Global
cache.metadata.jwks.expireAfterAccess	duration	Specifies the duration after a JWK is not accessed from the cache that it expires. By default this is disabled.	2m	Global
cache.metadata.jwks.expireAfterWrite	duration	Specifies the duration after a JWK is cached, that is, it expires and has to be loaded again. Default is 5 minutes.	5m	Global
db.invalidPoolTimeout	duration	Specifies how long to wait before retrying an invalid pool. Default: 15m	Not applicable	Global
db.poolDestroyTimeout	duration	Indicates how long to wait to gracefully destroy a pool before moving to forcefully destroy all connections including borrowed ones. Default: 5m	Not applicable	Pool specific

Table C-1 (Cont.) Oracle REST Data Services Configuration Settings

Key	Type	Description	Example	Setting Type
db.wallet.zip	string	Specifies the wallet archive (provided in BASE64 encoding) containing connection details for the pool.	Not applicable	Pool specific
db.wallet.zip.path	string	Specifies the path to a wallet archive containing connection details for the pool.	Not applicable	Pool specific
db.wallet.zip.service	string	Specifies the service name in the wallet archive for the pool.	Not applicable	Pool specific
debug.trackResources	boolean	Specifies to enable tracking of JDBC resources. If not released causes in resource leaks or exhaustion in the database. Tracking imposes a performance overhead.	Not applicable	Pool specific
feature.graphql.max.nesting.depth	numeric	Specifies the maximum join nesting depth limit for GraphQL queries. Defaults to 5.	10	Global
feature.openervicebroker.exclude	boolean	Specifies to disable the Open Service Broker services available for the pool.	Not applicable	Pool specific
feature.sdw	boolean	Specifies to enable the Database Actions feature.	Not applicable	Pool specific
http.cookie.filter	string	Specifies a comma separated list of HTTP Cookies to exclude when initializing an Oracle Web Agent environment.	Not applicable	Pool specific
jdbc.auth.admin.role	string	Identifies the database role that indicates that the database user must get the SQL Administrator role.	Not applicable	Pool specific
jdbc.cleanup.mode	Not applicable	Specifies how a pooled JDBC connection and corresponding database session, is released when a request has been processed. Default value is <code>RECYCLE</code>	Not applicable	Pool specific
owa.trace.sql	boolean	Specifies a boolean property. If it is true, then it causes a trace of the SQL statements performed by Oracle Web Agent to be echoed to the log.	Not applicable	Pool specific
plsql.gateway.mode	string	Indicates if the PL/SQL Gateway functionality should be available for a pool or not. Value can be one of <code>disabled</code> , <code>direct</code> , or <code>proxied</code> . <ul style="list-style-type: none"> If the value is <code>direct</code>, then the pool serves the PL/SQL Gateway requests directly. If the value is <code>PLSQL_GATEWAY_CONFIG</code>, view is used to determine the user to whom to proxy. 	proxied	Pool specific

Table C-1 (Cont.) Oracle REST Data Services Configuration Settings

Key	Type	Description	Example	Setting Type
request.traceHeaderName	string	Specifies the name of the HTTP request header that uniquely identifies the request end to end as it passes through the various layers of the application stack. In Oracle this header is commonly referred to as the ECID (Entity Context ID).	Not applicable	Global
resource.templates.enabled	boolean	Deprecated. Configuration property indicating if the legacy resource templates (APEX based REST) should be enabled or not. False by default. The resource-templates code base is not compatible with the single pool (ORDS_PUBLIC_USER) architecture so must be disabled.	Not applicable	Global
security.jwt.profile.enabled	boolean	Specifies whether the JWT Profile authentication is available. Supported values: <ul style="list-style-type: none"> true (default) false 	true	Pool specific
security.jwks.size	numeric	Specifies the maximum number of bytes read from the JWK url. Default 100000 bytes.	100000	Pool specific
security.jwks.connection.timeout	duration	Specifies the maximum amount of time before timing-out when accessing a JWK url. Default is 5 seconds.	5s	Pool specific
security.jwks.read.timeout	duration	Specifies the maximum amount of time reading a response from the JWK url before timing-out. Default is 5 seconds.	5s	Pool specific
security.jwks.refresh.interval	duration	Specifies the minimum interval between refreshing the JWK cached value.	10s	Pool specific
security.jwt.allowed.skew	duration	Specifies the maximum skew the JWT time claims are accepted. This is useful if the clock on the JWT issuer and ORDS differs by a few seconds. By default, it is disabled.	10s	Pool specific
security.jwt.allowed.age	duration	Specifies the maximum allowed age of a JWT in seconds, regardless of expired claim. The age of the JWT is taken from the JWT issued at claim. By default, it is disabled.	1h	Pool specific
security.credentials.attempts	numeric	Specifies the maximum number of unsuccessful password attempts allowed. Enabled by setting a positive integer value. Defaults to -1.	3	Global
security.credentials.file	string	Specifies the file where credentials are stored.	Not applicable	Global

Table C-1 (Cont.) Oracle REST Data Services Configuration Settings

Key	Type	Description	Example	Setting Type
security.credentials.lock.time	duration	Specifies the period to lock the account that has exceeded maximum attempts. Defaults to 10m (10 minutes)	15m	Global
security.validationFunctionType	string	Indicates the type of security.requestValidationFunction: javascript or plsql. Defaults to plsql.	Not applicable	Pool specific
standalone.access.log	string	Specifies the path to the folder to store HTTP request access logs. If not specified, then no access log is generated.	Not applicable	Global
standalone.binds	string	Specifies the comma separated list of host names or IP addresses to identify a specific network interface on which to listen. Default 0.0.0.0.	Not applicable	Global
standalone.context.path	string	Specifies the context path where ords is located. Defaults to /ords	Not applicable	Global
standalone.doc.root	string	Points to the location where static resources to be served under the / root server path are located.	Not applicable	Global
standalone.http.port	numeric	Specifies the HTTP listen port. Default: 8080	8777	Global
standalone.http.cert	string	Specifies the SSL certificate path. If you are providing the SSL certificate, then you must specify the certificate location.	Not applicable	Global
standalone.http.cert.key	string	Specifies the SSL certificate key path. If you are providing the SSL certificate, you must specify the certificate key location.	Not applicable	Global
standalone.http.hostname	string	Specifies the SSL certificate hostname.	Not applicable	Global
standalone.https.port	numeric	Specifies the HTTPS listen port. Default: 8443	Not applicable	Global
standalone.static.context.path	string	Specifies the Context path where APEX static resources are located. Default: /i	Not applicable	Global
standalone.static.path	string	Specifies the path to the folder containing static resources required by APEX.	Not applicable	Global
standalone.stop.timeout	duration	Specifies the period for Standalone Mode to wait until it is gracefully shutdown. Default: 10s (10 seconds)	15s	Global
apex.docTable	string	This parameter is deprecated, instead use owa.docTable parameter.	MYDOCTABLE	Pool specific

Table C-1 (Cont.) Oracle REST Data Services Configuration Settings

Key	Type	Description	Example	Setting Type
cache.metadata.timeout	string	Specifies the setting to determine for how long a metadata record remains in the cache. Longer duration means, it takes longer to view the applied changes. The formats accepted are based on the ISO-8601 duration format.	5m	Global
cache.metadata.enabled	boolean	Specifies the setting to enable or disable metadata caching. Supported values: <ul style="list-style-type: none"> true false (default) 	false	Global
database.api.enabled	boolean	Specifies whether the Database API is enabled. Supported values: <ul style="list-style-type: none"> true false (default) 	Not applicable	Global
db.connectionType	string	The type of connection. Supported values: <ul style="list-style-type: none"> basic tns customurl 	basic	Pool specific
db.customURL	string	Specifies the JDBC URL connection to connect to the database.	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=myhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=ora111.example.com)))	Pool specific
db.hostname	string	Specifies the host system for the Oracle database.	myhostname	Pool specific
db.password	string	Specifies the password of the specified database user. Include an exclamation at the beginning of the password so that it can be stored encrypted.	!password4user	Pool specific
db.port	numeric	Specifies the database listener port.	1521	Pool specific
db.servicename	string	Specifies the network service name of the database.	ora111.example.com	Pool specific

Table C-1 (Cont.) Oracle REST Data Services Configuration Settings

Key	Type	Description	Example	Setting Type
db.serviceNameSuffix	string	Specifies that the pool points to a CDB, and that the PDBs connected to that CDB should be made addressable by Oracle REST Data Services (see Making All PDBs Addressable by Oracle REST Data Services (Pluggable Mapping)).	apex_pu	Pool specific
db.sid	string	Specifies the name of the database.	ora111	Pool specific
db.tnsAliasName	string	Specifies the TNS alias name that matches the name in the tnsnames.ora file.	MY_TNSALIAS	Pool specific
db.tnsDirectory	string	The directory location of your tnsnames.ora file.	C:\ORACLE\NETWORK\ADMIN	Pool specific
db.username	string	Specifies the name of the database user for the connection.	APEX_PUBLIC_USER	Pool specific
debug.printDebugToScreen	boolean	Specifies whether to display error messages on the browser. Supported values: <ul style="list-style-type: none"> true false (default) 	false	Global
error.responseFormat	string	Specifies how the HTTP error responses must be formatted. Supported values: <ul style="list-style-type: none"> html - Force all responses to be in HTML format json - Force all responses to be in JSON format auto - Automatically determines most appropriate format for the request (default). 	json	Global
error.externalPath	string	Specifies the path to a folder that contains the custom error page.	/path/to/error/pages/folder/	Global
icap.port	numeric	Specifies the Internet Content Adaptation Protocol (ICAP) port to virus scan files. Either icap.port or icap.secure.port are required to have a value.	1344	Global
icap.secure.port	numeric	Specifies the Internet Content Adaptation Protocol (ICAP) port to virus scan files. Either icap.port or icap.secure.port are required to have a value. If values for both icap.port and icap.secure.port are provided, then the value of icap.port is ignored.	1344	Global

Table C-1 (Cont.) Oracle REST Data Services Configuration Settings

Key	Type	Description	Example	Setting Type
<code>icap.server</code>	string	Specifies the Internet Content Adaptation Protocol (ICAP) server name or IP address to virus scan files. The <code>icap.server</code> is required to have a value.	<code>servername</code>	Global
<code>jdbc.DriverType</code>	string	Specifies the JDBC driver type. Supported values: <ul style="list-style-type: none"> <code>thin</code> (Default) <code>oci8</code> 	<code>thin</code>	Pool specific
<code>jdbc.InactivityTimeout</code>	numeric	Specifies how long an available connection can remain idle before it is closed. The inactivity connection timeout is in seconds. Defaults to 1800.	1800	Pool specific
<code>jdbc.InitialLimit</code>	numeric	Specifies the initial size for the number of connections that will be created. Defaults to 10. (The default is low, and should probably be set higher in most production environments.)	10	Pool specific
<code>jdbc.MaxConnectionReuseCount</code>	numeric	Specifies the maximum number of times to reuse a connection before it is discarded and replaced with a new connection. Defaults to 1000.	1000	Pool specific
<code>jdbc.MaxLimit</code>	numeric	Specifies the maximum number of connections. Defaults to 10. (Might be too low for some production environments.)	20	Pool specific
<code>jdbc.auth.enabled</code>	boolean	Specifies if the PL/SQL Gateway calls can be authenticated using database users. If the value is <code>true</code> then this feature is enabled. If the value is <code>false</code> , then this feature is disabled. The default value is <code>false</code> . Oracle recommends not to use this feature. This feature used only to facilitate customers migrating from <code>mod_plsql</code> .	<code>false</code>	Pool specific
<code>jdbc.MaxStatementsLimit</code>	numeric	Specifies the maximum number of statements to cache for each connection. Defaults to 10.	10	Pool specific
<code>jdbc.MinLimit</code>	numeric	Specifies the minimum number of connections. Defaults to 2.	1	Pool specific

Table C-1 (Cont.) Oracle REST Data Services Configuration Settings

Key	Type	Description	Example	Setting Type
jdbc.statementTimeout	numeric	Specifies a timeout period on a statement. An abnormally long running query or script, executed by a request, may leave it in a hanging state unless a timeout is set on the statement. Setting a timeout on the statement ensures that all the queries automatically timeout if they are not completed within the specified time period. Defaults to 900.	900	Pool specific
log.procedure	boolean	Specifies whether procedures are to be logged. Supported values: <ul style="list-style-type: none"> true false (default) 	false	Global
misc.defaultPage	string	Specifies the default page to display. The Oracle REST Data Services Landing Page.	apex	Pool specific
misc.pagination.maxRows	numeric	Specifies the maximum number of rows that will be returned from a query when processing a RESTful service and that will be returned from a nested cursor in a result set. Affects all RESTful services generated through a SQL query, regardless of whether the resource is paginated. Defaults to 10000.	300	Pool specific
owa.docTable	string	Specifies the name of the document table used by the file upload. Defaults to FLOWS_FILES.WWV_FLOW_FILE_OBJECTS\$ value.	MYDOCTABLE	Pool specific

 **Note:**

For APEX 4.x and above this parameter should not be used.

Table C-1 (Cont.) Oracle REST Data Services Configuration Settings

Key	Type	Description	Example	Setting Type
<code>procedure.postProcess</code>	string	Specifies the procedure name(s) to execute after executing the procedure specified on the URL. Multiple procedure names must be separated by commas.	<code>SCHEMA1.SUBMIT.REQUEST, FINISHTASK</code>	Pool specific
<code>procedure.preProcess</code>	string	Specifies the procedure name(s) to execute prior to executing the procedure specified on the URL. Multiple procedure names must be separated by commas.	<code>SCOTT.PREPROC1, INITIALIZE, PKG1.PROC</code>	Pool specific
<code>procedure.request.preHook</code>	string	Specifies the function to be invoked prior to dispatching each Oracle REST Data Services based REST Service. The function can perform configuration of the database session, perform additional validation or authorization of the request. If the function returns <code>true</code> , then processing of the request continues. If the function returns <code>false</code> , then processing of the request is aborted and an HTTP 403 Forbidden status is returned.	<code>MYAPP.VALIDATE_REST_CALL</code>	Pool specific
<code>security.disableDefaultExclusionList</code>	boolean	<p>If this value is set to <code>true</code>, then the Oracle REST Data Services internal exclusion list is not enforced.</p> <p>Note: The Oracle REST Data Services internal exclusion list blocks the users from accessing the following:</p> <ul style="list-style-type: none"> • <code>sys.*</code> • <code>dbms_*</code> • <code>utl_*</code> • <code>owa_*</code> • <code>owa.*</code> • <code>htp.*</code> • <code>htf.*</code> • <code>wpg_docload.*</code> <p>Supported values:</p> <ul style="list-style-type: none"> • <code>true</code> • <code>false</code> (default) <p>Oracle recommends that you do not set this value to <code>true</code>. That is, do not disable the default internal exclusion list. The only possible exception is temporarily disabling the internal exclusion list for debugging purposes.</p>	<code>false</code>	Global

Table C-1 (Cont.) Oracle REST Data Services Configuration Settings

Key	Type	Description	Example	Setting Type
security.exclusionList	string	<p>Specifies a pattern for procedures, packages, or schema names which are forbidden to be directly executed from a browser.</p> <p>Procedure names can contain the wildcard characters asterisk (*) or question mark (?). Use an asterisk (*) to substitute zero or more characters and a question mark (?) to substitute for any one character.</p> <p>Note: Separate multiple patterns using commas.</p>	customer_accoun t,bank*, employee?	Global
security.inclusionList	string	<p>Specifies a pattern for procedures, packages, or schema names which are allowed to be directly executed from a browser.</p> <p>Procedure names can contain the wildcard characters asterisk (*) or question mark (?). Use an asterisk (*) to substitute zero or more characters and a question mark (?) to substitute for any one character.</p> <p>Note: Separate multiple patterns using commas.</p>	apex, p, v, f, wwv_*, y*, c*	Global
security.maxEntries	numeric	<p>Specifies the maximum number of cached procedure validations. Defaults to 2000. Set this value to 0 to force the validation procedure to be invoked on each request.</p>	2000	Global
security.requestAuthenticationFunction	string	<p>Specifies an authentication function to determine if the requested procedure in the URL should be allowed or disallowed for processing. The function should return <code>true</code> if the procedure is allowed; otherwise, it should return <code>false</code>. If it returns <code>false</code>, Oracle REST Data Services will return <code>WWW-Authenticate</code> in the response header.</p>	owa_custom.au thorize	Pool specific
security.requestValidationFunction	string	<p>Specifies a validation function to determine if the requested procedure in the URL should be allowed or disallowed for processing. The function should return <code>true</code> if the procedure is allowed; otherwise, return <code>false</code>.</p>	CHECK_VALID_P ROCEDURE	Pool specific

Table C-1 (Cont.) Oracle REST Data Services Configuration Settings

Key	Type	Description	Example	Setting Type
<code>security.verifySSL</code>	boolean	Specifies whether HTTPS is available in your environment. Supported values: <ul style="list-style-type: none"> • true (default) • false If you change the value to false, see Using OAuth2 in Non-HTTPS Environments .	<code>true</code>	Global
<code>soda.defaultLimit</code>	string	When using the SODA REST API, specifies the default number of documents returned for a GET request on a collection when a limit is not specified in the URL. Must be a positive integer, or "unlimited" for no limit. Defaults to 100.	<code>75</code>	Pool specific
<code>soda.maxLimit</code>	string	When using the SODA REST API, specifies the maximum number of documents that will be returned for a GET request on a collection URL, regardless of any limit specified in the URL. Must be a positive integer, or "unlimited" for no limit. Defaults to 1000.	<code>700</code>	Pool specific
<code>restEnabledSql.active</code>	boolean	Specifies whether the REST-Enabled SQL service is active. Supported values: <ul style="list-style-type: none"> • true • false (default) 	<code>true</code>	Pool specific

Index

A

Apache Tomcat, [1-2](#)
 about, [6-10](#)
 configuring Oracle REST Data Services
 images, [6-10](#)
 deploying to, [6-10](#)
 downloading, [6-10](#)

C

configuration file editable parameters
 jdbc.MaxConnectionReuseCount, [C-10](#)
 jdbc.MinLimit, [C-10](#)
 security.requestValidationFunction, [C-13](#)
 security.verifySSL, [C-14](#)
 soda.defaultLimit, [C-14](#)
 soda.maxLimit, [C-14](#)
configuration files, [C-1](#)
configuration folder
 structure of, [C-1](#)

D

database users, [1-5](#)
deploy options
 Apache Tomcat, [6-10](#)
deployment options
 Oracle WebLogic Server, [6-4](#)
DER
 converting private key to DER, [6-3](#)
downloading
 Apache Tomcat, [6-10](#)
 Oracle WebLogic Server, [6-5](#)

J

Java EE application servers
 about supported, [1-2](#)

M

multiple database configuration
 about the request URL, [3-1](#)
 configuring additional databases, [3-1](#)

O

OAuth2, default behavior, [5-15](#)
Oracle Cloud Infrastructure
 about, [6-12](#)
Oracle REST Data Services
 configuration files, [C-1](#)
 configuring, [5-1](#)
 database users, [1-5](#)
 Deploying, [6-1](#)
 system requirements, [1-1](#)
Oracle WebLogic Server, [1-2](#)
 about, [6-5](#)
 deploy to, [6-4](#)
 downloading, [6-5](#)
 installing, [6-4](#)
 installing the deployment, [6-7](#)

P

private key
 converting to DER, [6-3](#)

S

structure of configuration folder, [C-1](#)
supported Java EE application servers, [1-2](#)
system requirements, [1-1](#)