

Oracle® Retail Omnichannel Cloud Data Service

Installation Guide

Release 19.1.000

F32768-04

May 2021

Primary Author: Owen Horne

Contributing Author:

Contributor:

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 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 installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon 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, Opteron, the AMD logo, and the AMD Opteron 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.

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

- (i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.
- (ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.
- (iii) the software component known as **Access Via**[™] licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.
- (iv) the software component known as **Adobe Flex**[™] licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications.

Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

| | |
|---|-----|
| Send Us Your Comments | vii |
| Preface | ix |
| Audience | ix |
| Documentation Accessibility | ix |
| Related Documents | ix |
| Customer Support | ix |
| Review Patch Documentation | x |
| Improved Process for Oracle Retail Documentation Corrections | x |
| Oracle Retail Documentation on the Oracle Help Center (docs.oracle.com) | x |
| Conventions | x |
| | |
| 1 Introduction | |
| OCDS Topology | 1-2 |
| | |
| 2 Technical Specifications | |
| Requesting Infrastructure Software | 2-1 |
| Server Requirements | 2-1 |
| Installation Sequence | 2-1 |
| Software Dependencies | 2-1 |
| Oracle Retail Product Compatibility Matrix | 2-2 |
| | |
| 3 OCDS Schemas | |
| Prerequisites | 3-1 |
| Preparation | 3-2 |
| Database Schema Population | 3-2 |
| Enable REST Services on OCDS Database | 3-2 |
| Secure OCDS Web Services on OCDS Database | 3-3 |
| | |
| 4 WebLogic Middleware | |
| Installing WebLogic | 4-1 |
| Creating Schemas with the Repository Creation Utility (RCU) | 4-1 |
| Creating a WebLogic Domain with JRF | 4-1 |
| Prerequisites | 4-1 |

| | |
|--|-----|
| WebLogic Domain Creation..... | 4-2 |
| 5 OCDS (BDI) Job Admin | |
| Prerequisites..... | 5-1 |
| Preparation..... | 5-1 |
| Job Admin Installation..... | 5-3 |
| Verify Installation..... | 5-5 |
| 6 OCDS (RIB) Injector | |
| Prerequisites..... | 6-1 |
| Preparation..... | 6-1 |
| Injector Installation..... | 6-2 |
| Verify Installation..... | 6-3 |
| 7 OCDS (ORDS) Web Services | |
| Prerequisites..... | 7-1 |
| Preparation..... | 7-1 |
| Deploy ORDS..... | 7-1 |
| Verify Installation..... | 7-3 |
| A Appendix A: Migrating OCDS to v19.1 | |
| Preliminary Requirements..... | A-1 |
| OCDS Schema Upgrade..... | A-1 |
| Pre Schema Upgrade..... | A-1 |
| OCDS Schema Migration..... | A-1 |
| Post Schema Upgrade..... | A-2 |
| WAR FILE Upgrades | A-2 |
| OCDS jobAdmin.warUpgrade..... | A-2 |
| Edit Deployment Configuration File..... | A-2 |
| Job Admin Deployment..... | A-4 |
| OCDS ocds-injector.war Upgrade..... | A-4 |
| Preparation..... | A-5 |
| Edit Deployment Configuration Files..... | A-5 |
| OCDS Injector Deployment..... | A-6 |
| ORDS Upgrade..... | A-6 |

Send Us Your Comments

Oracle® Retail Omnichannel Cloud Data Service Installation Guide, 19.1.000

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Help Center (docs.oracle.com) Web site. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at <http://www.oracle.com>.

Preface

The *Oracle® Retail Omnichannel Cloud Data Service Installation Guide* provides information about the processing of the Oracle Omnichannel Cloud Data Service (OCDS) data hub.

Audience

This guide is for technical personnel who configure, maintain and support, or use Oracle Retail Omnichannel Cloud Data Service.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the Oracle Retail documentation set.

<https://docs.oracle.com/en/industries/retail/index.html>

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

<https://support.oracle.com>

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received

- Screen shots of each step you take

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 19.1.000) or a later patch release (for example, 19.1.001). If you are installing the base release or additional patches, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times not be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Help Center (docs.oracle.com) Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available on the Retail landing page at the Oracle Help Center (OHC) at the following URL:

<https://docs.oracle.com/en/industries/retail/index.html>

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

Oracle Retail Documentation on the Oracle Help Center (docs.oracle.com)

Oracle Retail product documentation is also available on the following web site:

<https://docs.oracle.com/en/industries/retail/index.html>

(Data Model documents can be obtained through My Oracle Support.)

Conventions

The following text conventions are used in this document:

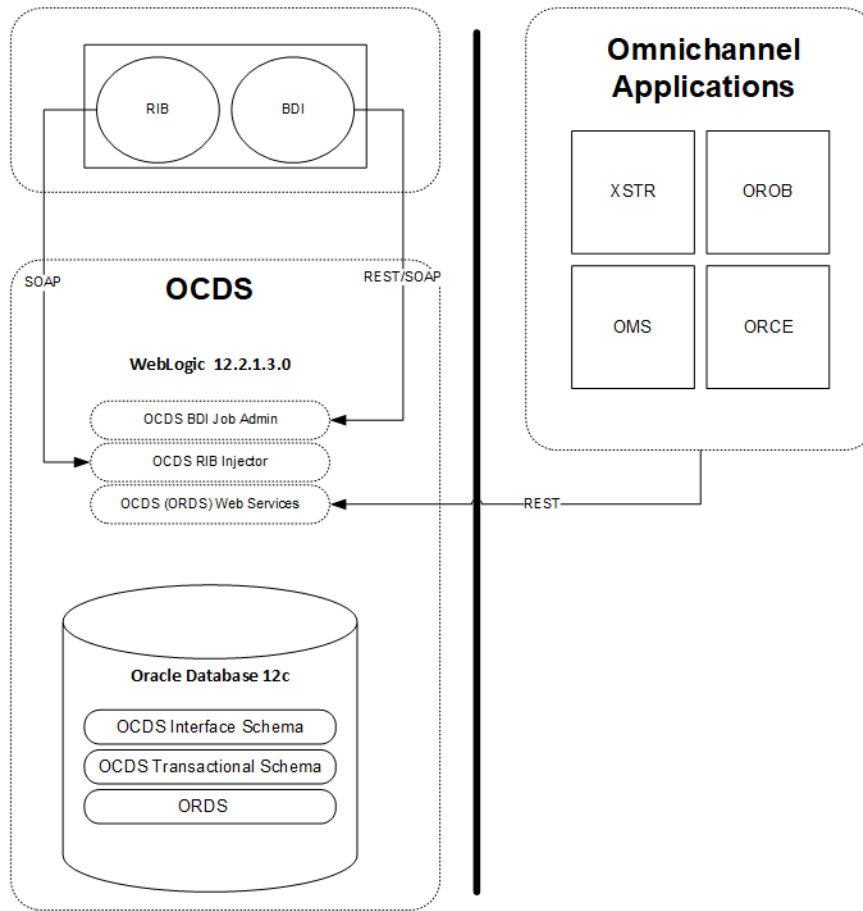
| Convention | Meaning |
|-----------------|--|
| boldface | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary. |
| <i>italic</i> | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular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

Introduction

Oracle Omnichannel Cloud Data Service (OCDS) is a data hub, enabling Oracle Retail Merchandising and Pricing applications to share foundation data with Oracle Retail Omnichannel applications. OCDS contains the following components:

- BDI Batch Job Admin - Enables in-bound data flow into OCDS using Oracle Bulk Data Integration (BDI) technology. Job Admin has a User Interface (UI) to support the management of BDI batch Jobs.
- RIB Injector - Enables in-bound data flow into OCDS from the Oracle Retail Integration Bus (RIB).
- ORDS - Enables out-bound data flow from OCDS to Omnichannel Applications through the use of RESTful web services.

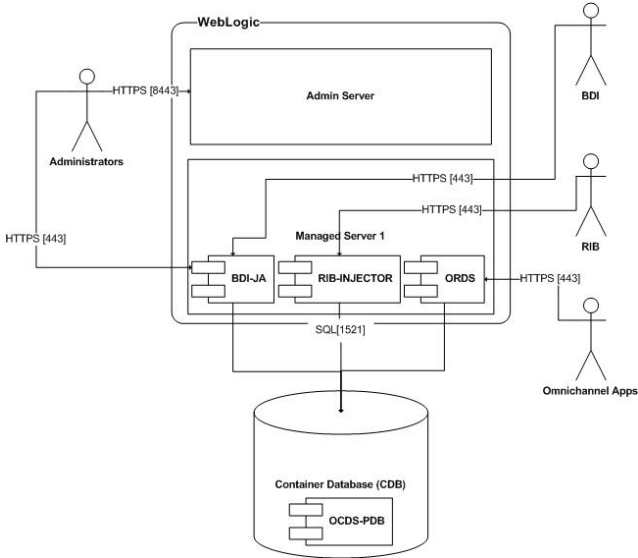
Figure 1–1 OCDS Components



OCDS Topology

The diagram below illustrates the basic deployment topology for OCDS. Alternatively, each OCDS component can be hosted in its own WebLogic Managed Server.

Figure 1-2 Basic Deployment



- **BDI-JA:** OCDS (BDI) Job Admin is the interface between the Oracle Retail Bulk Data Integration and OCDS, enabling BDI data to flow into the OCDS database.
- **RIB-INJECTOR:** OCDS (RIB) Injector is the interface between RIB infrastructure and OCDS; it listens for SOAP-based RIB messages containing incremental changes to data initially populated through BDI.
- **ORDS:** The OCDS (ORDS) Web Service exposes the data managed by OCDS to Omnichannel applications.

Technical Specifications

Oracle Omnichannel Cloud Data Service (OCDS) has several dependencies. This section covers these requirements.

Requesting Infrastructure Software

If you are unable to find the necessary version of the required Oracle infrastructure software (database server, application server, WebLogic, and so on) on the Oracle Software Delivery Cloud, you should file a non-technical Contact Us Service Request (SR) and request access to the media. For instructions on filing a non-technical SR, see *My Oracle Support Note 1071023.1 - Requesting Physical Shipment or Download URL for Software Media*.

Server Requirements

For more information on server requirements, see the *Oracle Retail Bulk Data Integration Cloud Service Installation Guide*.

Installation Sequence

It is recommended that the installation of OCDS is performed in the order presented in this guide.

1. Create OCDS Schemas.
2. Create a WebLogic Domain.
3. The following OCDS components can be installed and deployed in any order:
 - Install and deploy OCDS (BDI) Job Admin.
 - Install and deploy OCDS (RIB) Injector.
 - Install and deploy OCDS (ORDS) Web Services.

Software Dependencies

The installation and operation of Oracle Omnichannel Cloud Data Service (OCDS) depends several Oracle and third-party software, in addition to the OCDS distribution files. The following should be performed before starting the OCDS install process.

- Install Java JDK 8 or later.
- Install Oracle Database 12c (Release 12.1.0.2).
- Download Oracle Fusion Middleware (WebLogic 12.2.1.3.0).

- Download Oracle REST Data Services 19.2 (ords-19.2.0.199.1647zip).
<https://www.oracle.com/database/technologies/appdev/rest-data-services-v192-downloads.html>

If upgrading from a previous Oracle REST Data Service follow the instructions provided in the download package.

Oracle Retail Product Compatibility Matrix

The table shows the certified OCDS version compatible with the Merchandising Suite version.

Table 2-1 Oracle Retail Product Compatibility Matrix

| Products | OCDS Version | Merchandising Suite Version |
|--------------------------|---------------------|------------------------------------|
| OCDS/Merchandising Suite | 19.1.002 | 19.2.001 |

OCDS Schemas

This chapter describes the instructions for building the OCDS schemas on an Oracle 12c Pluggable Database (PDB).

Prerequisites

1. Oracle Database 19c (Release 19.3.0.0.0) has been installed.
2. Container Database (CDB) has been created.
3. Pluggable Database (PDB) for OCDS schema has been created.
4. Configured ORDS 19.2 for the OCDS database:

Note: For information on the ORDS version recommended for Oracle Database 19c, see the Oracle REST Data Services documentation on the Oracle Help Center (docs.oracle.com) at:

<https://docs.oracle.com/en/database/oracle/oracle-rest-data-services/>

ORDS should be installed AND validated prior to any OCDS installation.

- Set the location of the ORDS configuration files
`java -jar ords.war configdir </path/to/ords/config>`
 - Configure database connection to the OCDS database
`java -jar ords.war setup --database <database name>`
 - Configure the request routing rule for OCDS services
`java -jar ords.war map-url --type base-path <path prefix>
<database name>`
5. The two OCDS database users have been created with the following names and empty schemas:
 - `ocds_ifc`
 - `ocds_txn`

Preparation

Perform the following procedure to prepare for these schema creation of the OCDS database. This archive file contains scripts to populate the two OCDS schemas, enable and secure the OCDS REST services.

- Unzip `ocds-database-creation.zip`. The location where the files were extracted will be referenced as `<dbScripts>` in the following steps.

Database Schema Population

Perform the following steps to populate the OCDS schemas.

1. Connect to the interface `ocds_ifc` schema and populate it. The interface schema needs to be complete prior to proceeding with step 2.

Note: Starting with patch release 19.1.002 you only need to run the `<dbScripts>/scripts/rtg_ifc/ddl/bdi_ocds_ddl.sql` script. You do not need to run the `<dbScripts>/scripts/rtg_ifc/ddl/BDI_PUBLIC_INTERFACE_TABLE_CREATE.sql` script.

- a. `<dbScripts>/scripts/rtg_ifc/ddl/BDI_PUBLIC_INTERFACE_TABLE_CREATE.sql`
- b. `<dbScripts>/scripts/rtg_ifc/ddl/bdi_ocds_ddl.sql`

See the *Oracle Retail Bulk Data Integration Cloud Service Installation Guide*, for more information on how to create the interface schema or to troubleshoot issues.

2. Connect to the `ocds_ifc` schema as a user with permissions to grant access to tables in the `ocds_ifc` schema and execute the following scripts:
 - `<dbScripts>/ocds_txn/plsql/Interface_Schema_Access.sql`
3. Connect to the `ocds_txn` schema and execute the following scripts:
 - `<dbScripts>/scripts/ocds_txn/ddl/ocds-txn-ddl.sql`
 - `<dbScripts>/scripts/ocds_txn/plsql/ocds-txn-plsql.sql`
4. Connect to the `ocds_txn` schema as a user with permissions to grant access to packages on the `ocds_txn` schema and execute the following scripts:
 - `<dbScripts>/scripts/ocds_ifc/plsql/Transaction_Schema_Access.sql`
5. Connect to the `ocds_ifc` schema and execute the following scripts:
 - `<dbScripts>/scripts/ocds_ifc/plsql/ocds-ifc-plsql.sql`

Enable REST Services on OCDS Database

Perform the following procedure to enable the OCDS web services on the `ocds_txn` schema.

1. Connect to the `ocds_txn` schema and execute the following script:
 - `<dbScriptRoot>/scripts/ocds_txn/rest/ocds-enable-rest.sql`

Secure OCDS Web Services on OCDS Database

Perform the following procedure to secure the OCDS web services on the `ocds_txn` schema.

1. Connect to the `ocds_txn` schema and execute the following script:
 - `<dbScriptRoot>/scripts/ocds_txn/rest/SecureRest.sql`

WebLogic Middleware

This chapter describes the procedure for installing and creating the WebLogic Middleware needed to host OCDS. Important information about the installation and deployment of a BDI Job Admin can be found in the *Oracle Retail Bulk Data Integration Installation Guide*.

Installing WebLogic

For more information about installing WebLogic, see the *Oracle Retail Bulk Data Integration Cloud Service Installation Guide*.

Creating Schemas with the Repository Creation Utility (RCU)

The installation of OCDS Job Admin and Injector components requires the existence of schemas in a database prior to installation. These schemas are created and loaded in your database using the Repository Creation Utility (RCU).

For more information on how to create the required schema using the Repository Creation Utility, see the *Oracle Retail Bulk Data Integration Cloud Service Installation Guide*.

Creating a WebLogic Domain with JRF

This section describes instructions for creating a new WebLogic domain with JRF, and instructions to create a managed server into which the OCDS Job Admin, Injector, and ORDS components can be deployed.

Prerequisites

The installation of OCDS components requires the existence of schemas in a database prior to installation. These schemas are created and loaded in your database using the Repository Creation Utility (RCU). OCDS requires Oracle WebLogic server 12c (12.2.1.3.0), built with Java 8 (JDK 1.8 64 bit with the latest security updates).

The minimum recommended Java VM memory setting for the OCDS application domain is:

```
-Xms1024m -Xmx2048m
```

If re-creating a domain using the same RCU schemas, and those schemas are not in `ocds_*` tablespaces, then run RCU to drop old RCU schemas.

WebLogic Domain Creation

For more information on how to create a WebLogic domain with JRF, see the *Oracle Retail Bulk Data Integration Cloud Service Installation Guide*.

OCDS (BDI) Job Admin

This chapter describes the procedure to install and deploy the OCDS (BDI) Job Admin application on a WebLogic domain. The OCDS (BDI) Job Admin is an Oracle Retail Bulk Data Integration component. Additional information can be found about the Installation of a BDI Job Admin in the *Oracle Retail Bulk Data Integration Installation Guide*.

Prerequisites

The installation of OCDS Job Admin component requires the existence of schemas in a database prior to installation. These schemas are created and loaded in your database using the Repository Creation Utility (RCU) described in the previous section, and the steps outlined in the OCDS Schemas chapter of this document.

The target WebLogic Admin Server and Managed Server should be running.

The `JAVA_HOME` environment variable must be set.

Preparation

Perform the following procedure to install the OCDS (BDI) Job Admin Application:

1. Unzip `ocds-jobadmin-deployment.zip`.
2. Configure the `conf/bdi-job-admin-deployment-env-info.json` file with the database and WebLogic domain details. This file is used by the deployment script.
 - a. Edit the Datasource definitions for `JobAdminDatasource`:
 - `jdbcUrl`: Configure the `jdbcUrl` for all `DataSources` definitions in `DataSourceDef`.

`BatchInfraDataSource` references a schema created using the WebLogic RCU (`<prefix>_WLS`).

All other datasources reference the OCDS interface (`ocds_ifc`) schema created during the prerequisite step: OCDS Database Creation.

Figure 5–1 Datasource Definitions

```

1 "BdiJobAdminDeploymentEnvInfo": {
  "DataSourceDef": {
    "JobAdminDataSource": {
      "dataSourceName": "OodsJobAdminDataSource",
      "dataSourceClass": "oracle.jdbc.pool.OracleDataSource",
      "dataSourceJndiName": "jdbc/OodsJobAdminDataSource",
      "jdbcUrl": "jdbc:oracle:thin://:1521/ocdspdb",
      "jdbcUserAlias": "ocdsJobAdminDataSourceUserAlias",
      "jdbcUser": "GET_FROM_WALLET",
      "jdbcPassword": "GET_FROM_WALLET",
      "dataSourceProperties": {
        "connectionPool_MaxCapacity": "300"
      }
    },
    "ReceiverServiceDataSource": {
      "dataSourceName": "OodsReceiverServiceDataSource",
      "dataSourceClass": "oracle.jdbc.pool.OracleDataSource",
      "dataSourceJndiName": "jdbc/OodsReceiverServiceDataSource",
      "jdbcUrl": "jdbc:oracle:thin://:1521/ocdspdb",
      "jdbcUserAlias": "ocdsReceiverServiceDataSourceUserAlias",
      "jdbcUser": "GET_FROM_WALLET",
      "jdbcPassword": "GET_FROM_WALLET",
      "dataSourceProperties": {
        "connectionPool_MaxCapacity": "300"
      }
    },
    "BatchInfraDataSource": {
      "dataSourceName": "BatchInfraDataSource",
      "dataSourceClass": "oracle.jdbc.xa.client.OracleXADataSource",
      "dataSourceJndiName": "jdbc/BatchInfraDataSource",
      "jdbcUrl": "jdbc:oracle:thin://:1521/ocdspdb",
      "jdbcUserAlias": "batchInfraDataSourceUserAlias",
      "jdbcUser": "GET_FROM_WALLET",
      "jdbcPassword": "GET_FROM_WALLET",
      "dataSourceProperties": {
        "connectionPool_MaxCapacity": "300"
      }
    },
    "JobXmlDataSource": {
      "dataSourceName": "JobXmlDataSource",
      "dataSourceClass": "oracle.jdbc.xa.client.OracleXADataSource",
      "dataSourceJndiName": "jdbc/JobXmlDataSource",
      "jdbcUrl": "jdbc:oracle:thin://:1521/ocdspdb",
      "jdbcUserAlias": "jobXmlDataSourceUserAlias",
      "jdbcUser": "GET_FROM_WALLET",
      "jdbcPassword": "GET_FROM_WALLET",
      "dataSourceProperties": {
        "connectionPool_MaxCapacity": "300"
      }
    }
  }
},
}

```

b. Edit the Middleware Server definitions for JobAdminAppServer

- webLogicDomainName: WebLogic domain name.
- webLogicDomainHome: WebLogic domain home directory.
- webLogicDomainAdminServerUrl: Server URL information.
- webLogicDomainAdminServerHost: Server host.
- webLogicDomainAdminServerPort: Admin Server port.
- webLogicDomainTargetManagedServerName: Managed Server name.
- jobAdminUiUrl: Host and managed server port where Job Admin application will be deployed. This can be setup with the HTTPS port.

Figure 5–2 OCDS Setup HTTPS Port

```

^ "MiddlewareServerDef":{
  "JobAdminAppServer": {
    "weblogicDomainName": "ocds_domain",
    "weblogicDomainHome": "F:\01\webadmin\products\wls_ocds/domains/ocds_domain",
    "weblogicDomainAdminServerUrl": "t3://localhost:8440",
    "weblogicDomainAdminServerProtocol": "t3",
    "weblogicDomainAdminServerHost": "localhost",
    "weblogicDomainAdminServerPort": "8440",
    "weblogicDomainAdminServerUserAlias": "OcdsAppServerAdminServerUserAlias",
    "weblogicDomainTargetManagedServerName": "OCDS_ManagedServer_1",

    "jobAdminUiUrl": "http://localhost:8442/ocds-batch-job-admin",
    "jobAdminUiUserGroup": "BdiEdgeOcdsJobAdminGroup",
    "jobAdminUiUserAlias": "ocdsJobAdminUiUserAlias",
    "jobAdminUiUser": "GET_FROM_WALLET",
    "jobAdminUiPassword": "GET_FROM_WALLET",

    "jobOperatorUiUserGroup": "BdiEdgeOcdsJobOperatorGroup",
    "jobOperatorUiUserAlias": "ocdsJobOperatorUiUserAlias",
    "jobOperatorUiUser": "GET_FROM_WALLET",
    "jobOperatorUiPassword": "GET_FROM_WALLET",

    "jobMonitorUiUserGroup": "BdiEdgeOcdsJobMonitorGroup",
    "jobMonitorUiUserAlias": "ocdsJobMonitorUiUserAlias",
    "jobMonitorUiUser": "GET_FROM_WALLET",
    "jobMonitorUiPassword": "GET_FROM_WALLET"
  }
},
"JobAdminApplication":{
  "appName": "ocds",
  "JobAdminAppUses": [
    "JobAdminDataSource",
    "JobAdminAppServer",
    {
      "RemoteJobAdminAppServers": []
    }
  ]
}
}

```

c. Edit RMS JobAdmin Server.

- - jobAdminUiUrl: Host and managed server port where Job Admin application will be deployed. This can be setup with the HTTPS port.

Figure 5–3 RMS JobAdmin Server Setup

```

"RmsJobAdminAppServer": {
  "jobAdminUiUrl": "http://localhost:7001/rms-batch-job-admin",
  "jobAdminUiUserAlias": "rmsJobAdminBaseUrUserAlias",
  "jobAdminUiUser": "GET_FROM_WALLET",
  "jobAdminUiPassword": "GET_FROM_WALLET",
}

```

Job Admin Installation

Perform the following procedure to install and deploy the Job Admin Application.

1. Change to the `ocds-jobadmin-deployment/bin` folder and execute the version `bdi-job-admin-deployer` script for the o/s using the switches:

```
-setup-credentials -deploy-job-admin-app
```

On Linux:

```
./bdi-job-admin-deployer.sh -setup-credentials -deploy-job-admin-app
```

On Windows:

```
bdi-job-admin-deployer.cmd -setup-credentials -deploy-job-admin-app
```

- a. There will be one prompt for a WebLogic user credential:

- Enter username for alias (`OcdsAppServerAdminServerUserAlias`):
Enter the WebLogic Admin Server credentials.

Figure 5–4 OCDS App Servers Admin Server User Alias

```

bash-4.2$ ./bdi-job-admin-deployer.sh -setup-credentials -deploy-job-admin-app
log4j:WARN No appenders could be found for logger (com.oracle.retail.integration.common.security.credential.CredentialStoreManager).
log4j:WARN Please initialize the log4j system properly.

Credential required for weblogicDomainAdminServerHost(localhost) weblogicDomainAdminServerPort(8440):
Enter username for alias (OcdsAppServerAdminServerUserAlias):weblogic
Enter Password:

```

- b. There will be three prompts to create JobAdmin user credentials:
- Enter username for alias (ocdsJobAdminUiUserAlias):
Enter credentials to be used to create the *Admin* user.
 - Enter username for alias (ocdsJobOperatorUiUserAlias):
Enter credentials to be used to create the *Operator* user.
 - Enter username for alias (ocdsJobMonitorUiUserAlias):
Enter credentials to be used to create the *Monitor* user.

Figure 5–5 Prompts to Create JobAdmin User Credentials

```

Credential required for jobAdminUiUrl(http://localhost:8442/ocds-batch-job-admin):
Enter username for alias (ocdsJobAdminUiUserAlias):ocdsadmin
Enter Password:
Prepare to use DB store for runtime credentials
Preparing to store Runtime credentials on the DB store with appTag (ocds-batch-job-admin.war)
Persisting runtime credentials to DB store

Credential required for jobOperatorUiUrl(http://localhost:8442/ocds-batch-job-admin):
Enter username for alias (ocdsJobOperatorUiUserAlias):ocdsoperator
Enter Password:
Persisting runtime credentials to DB store

Credential required for jobMonitorUiUrl(http://localhost:8442/ocds-batch-job-admin):
Enter username for alias (ocdsJobMonitorUiUserAlias):ocdsmonitor
Enter Password:

```

- c. There will be four prompts for database user credentials. Three of the four credentials are for the OCDS Interface User named `ocds_ifc`.
- Enter username for alias (ocdsJobAdminDataSourceUserAlias):
Enter the credentials for the OCDS Interface schema user. The username must be `ocds_ifc`. The password was defined as a prerequisite in the [Chapter 3, "OCDS Schemas"](#).
 - Enter username for alias (ocdsReceiverServiceDataSourceUserAlias):
Enter the credentials for the OCDS Interface schema user. The username must be `ocds_ifc`. The password was defined as a prerequisite in the [Chapter 3, "OCDS Schemas"](#).
 - Enter username for alias (batchInfraDataSourceUserAlias):
Enter the credentials for the `<prefix>_WLS` schema created during the Repository Creation Utility (RCU) step.

Figure 5–6 Prompts for Database User Credentials

```

Credential required for dataSource(jdbc/OcdeJobAdminDataSource) jdbcUrl(jdbc:oracle:thin://      :1521/ocdspdb):
Enter username for alias (ocdeJobAdminDataSourceUserAlias):ocds_ifc
Enter Password:

Credential required for ReceiverService dataSource(jdbc/OcdeReceiverServiceDataSource) jdbcUrl(jdbc:oracle:thin://      :1
521/ocdspdb):
Enter username for alias (ocdeReceiverServiceDataSourceUserAlias):ocds_ifc
Enter Password:

Credential required for BatchInfraDataSource dataSource(jdbc/BatchInfraDataSource) jdbcUrl(jdbc:oracle:thin://      :1521/
ocdspdb):
Enter username for alias (batchInfraDataSourceUserAlias):OCDS_WLS
Enter Password:

Credential required for JobXmlDataSource dataSource(jdbc/JobXmlDataSource) jdbcUrl(jdbc:oracle:thin://      :1521/ocdspdb):
Enter username for alias (jobXmlDataSourceUserAlias):ocds_ifc
Enter Password:

```

Verify Installation

After the OCDS (BDI) Job Admin application has been successfully deployed you should be able to access and log into the application's user interface.

1. Verify that the BDI Job Admin has been deployed.
 - a. Go to `http[s]://<host>:<port>/ocds-batch-job-admin/`
Example: `https://example:8443/ocds-batch-job-admin/`
 - b. At the prompt enter one of the Job Admin User credentials created during the installation.

Figure 5–7 Job Admin User Credentials

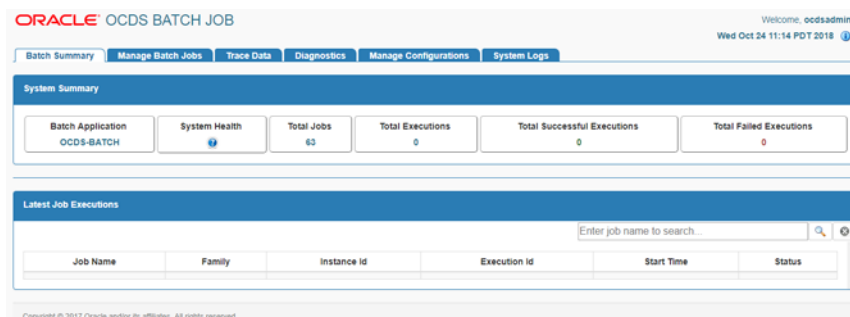
Sign in

`https://` :8443

Username

Password

- c. The OCDS Job Admin UI displays.

Figure 5–8 OCDS Job Admin UI


OCDS (RIB) Injector

This chapter describes the procedure to install and deploy the OCDS (RIB) Injector application on a WebLogic domain.

Prerequisites

The target WebLogic Admin Server and Managed Server should be running.

The `JAVA_HOME` environment variable must be set.

Preparation

Perform the following procedure to install the OCDS (RIB) Injector Application:

1. Configure the `conf/bdi-job-admin-deployment-env-info.json` file with the database and WebLogic domain details. This file is used by the deployment script.
 - a. Edit the Datasource definitions for `InjectorDataSource`.
 - `jdbcUrl`: This is the jdbc URL needed to connect to the OCDS Transactional schema. The OCDS Transactional (`ocds_txn`) schema was created during the prerequisite step: OCDS Database: Database Creation.

Figure 6–1 *jdbc URL*

```
"InjectorDeploymentEnvInfo": {
  "DataSourceDef": {
    "InjectorDataSource": {
      "dataSourceName": "InjectorDataSource",
      "dataSourceClass": "oracle.jdbc.pool.OracleDataSource",
      "dataSourceJndiName": "jdbc/InjectorDataSource",
      "jdbcUrl": "jdbc:oracle:thin:@//:1521/ocdspdb",
      "jdbcUserAlias": "InjectorDataSourceUserAlias",
      "jdbcUser": "GET_FROM_WALLET",
      "jdbcPassword": "GET_FROM_WALLET"
    }
  }
},
```

- b. Edit the Middleware Server definitions for `InjectorAppServer`.
 - `webLogicDomainName`: WebLogic domain name.
 - `webLogicDomainHome`: WebLogic domain home directory.
 - `webLogicDomainAdminServerUrl`: Server URL information.
 - `webLogicDomainAdminServerHost`: Server host.
 - `webLogicDomainAdminServerPort`: Admin Server port.
 - `webLogicDomainTargetManagedServerName`: Managed Server name.

Figure 6–2 Middleware Server Definitions

```

"MiddlewareServerDef":{
  "injectorAppServer": {
    "weblogicDomainName": "ocds_domain",
    "weblogicDomainHome": "/u017",
    "weblogicDomainAdminServerUrl": "t3://localhost:8440",
    "weblogicDomainAdminServerProtocol": "t3",
    "weblogicDomainAdminServerHost": "localhost",
    "weblogicDomainAdminServerPort": "8440",
    "weblogicDomainAdminServerUserAlias": "OcdsAppServerAdminServerUserAlias",
    "weblogicDomainTargetManagedServerName": "OCDS_ManagedServer_1",

    "injectorIntegrationUserGroup": "IntegrationGroup",
    "injectorIntegrationUserAlias": "IntegrationUserAlias",
    "injectorIntegrationUser": "GET_FROM_WALLET",
    "injectorIntegrationPassword": "GET_FROM_WALLET",
  },
},

```

Injector Installation

Perform the following procedures to install and deploy the Injector application.

1. Change to the `ocds-injector-deployment/bin` folder and execute the version of `injector-deployer` script for the o/s using the switches:

```
-setup-credentials -deploy-injector-app
```

On Linux:

```
./injector-deployer.sh -setup-credentials -deploy-injector-app
```

On Windows:

```
injector-deployer.cmd -setup-credentials -deploy-injector-app
```

- a. There will be one prompt for WebLogic user credentials:
 - Enter username for alias (`OcdsAppServerAdminServerUserAlias`):
Enter the WebLogic Admin Server credentials.

Figure 6–3 WebLogic User Credentials

```

bash-4.2$ ./injector-deployer.sh -setup-credentials -deploy-injector-app
Extracting jars from jps-wallet-all.
log4j:WARN No appenders could be found for logger (com.oracle.retail.integration.common.security.credential.CredentialStoreManager).
log4j:WARN Please initialize the log4j system properly.

Credential required for weblogicDomainAdminServerHost(localhost) weblogicDomainAdminServerPort(8440):
Enter username for alias (OcdsAppServerAdminServerUserAlias):weblogic
Enter Password:

```

- b. There will be one prompt to create the Integration User:
 - Enter username for alias (`IntegrationUserAlias`):
Enter credentials for the integration user. These credentials will enable RIB to communicate with OCDS.

Note: Password must not start with a number.

Figure 6–4 Integration User

```

Credential required for Integration User:
Enter username for alias (IntegrationUserAlias):integrationUser
Enter Password:

```

- c. There will be one prompt for database user credentials.

- Enter username for alias (InjectorDataSourceUserAlias):

Enter the credentials for the OCDS Transactional schema user. The username must be `ocds_txn`. The password was defined as a prerequisite in [Chapter 3, "OCDS Schemas"](#).

Figure 6–5 Prompt Database User Credentials

```
Credential required for dataSource(jdbc/InjectorDataSource) jdbcUrl(jdbc:oracle:thin://:1521/ocdspdb):
Enter username for alias (InjectorDataSourceUserAlias):ocds_txn
Enter Password:
```

Verify Installation

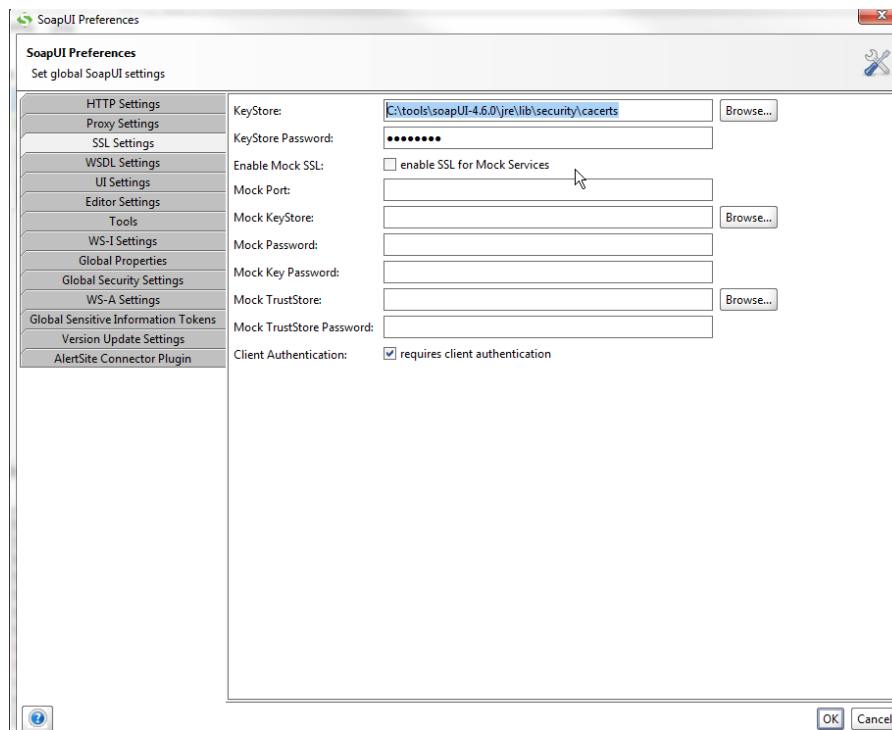
If the OCDS (RIB) Injector application has been successfully deployed then you should be able to verify the application is reported with an OK health status, and invoke a SOAP Web Service call from a tool like SOAP UI.

1. Verify the OCDS Injector Application (`injector.war`) is deployed and has a status of Active on the WLS Console.
2. The injector deployment can be more thoroughly verified by using the SOAP UI (<http://www.soapui.org>). Out of the box, the Injector is secured with RGPU PolicyA.

To configure SOAP UI to make SOAP requests:

- a. Add trusted SSL certificate to SOAPUI truststore. See SOAPUI preferences for location of truststore.

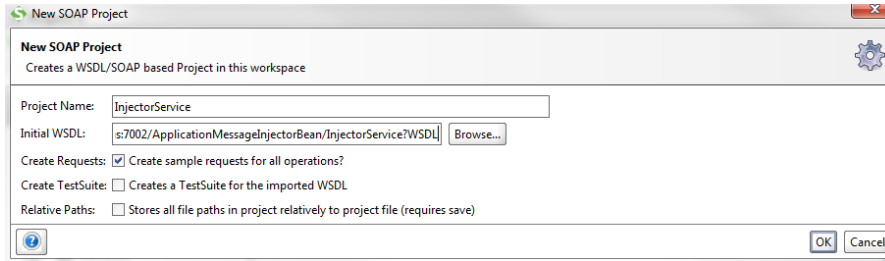
Figure 6–6 SOAP UI Preferences



- b. Create a new SOAP Project.

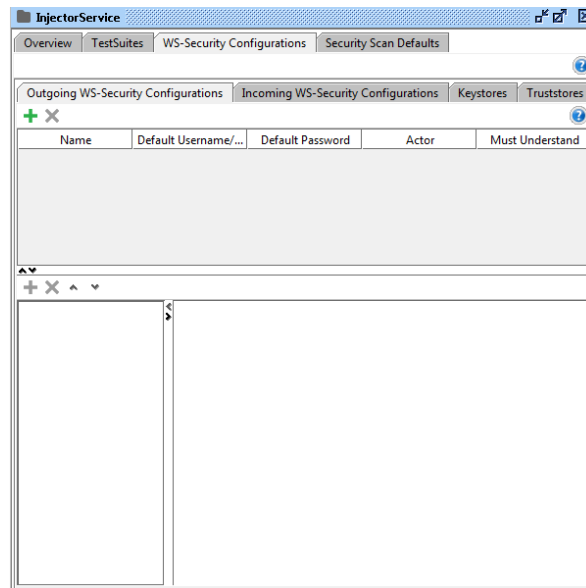
The WSDL location is `https://<host>:<port>/ApplicationMessageInjectorBean/InjectorService?WSDL`.

Figure 6–7 SOAP Project



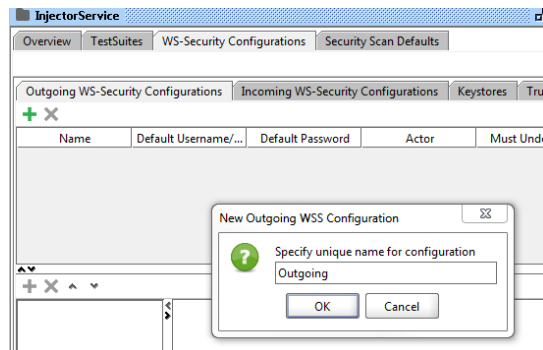
- c. Create an outgoing WS-Security Configuration (from Show Project View).

Figure 6–8 WS-Security Configuration



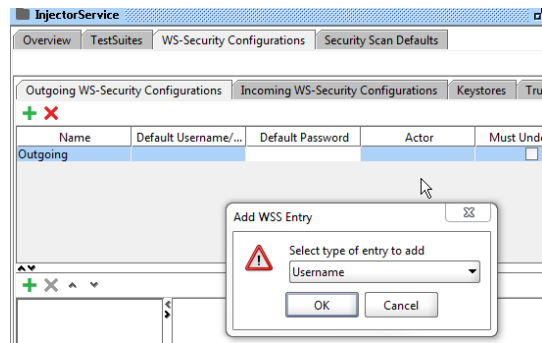
- d. Click the Plus sign to specify a unique name.

Figure 6–9 Name for Configuration



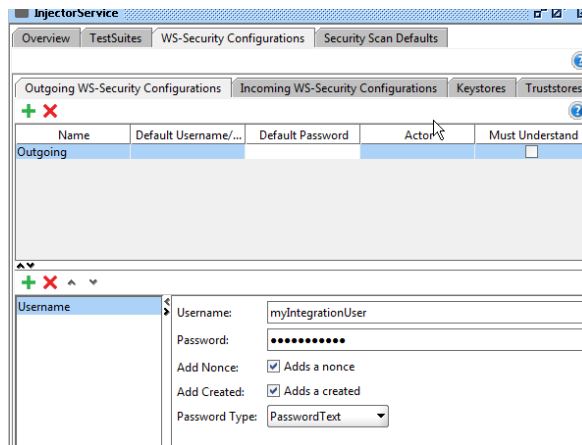
- e. Click the Plus sign in lower section to add user name WSS Entry.

Figure 6–10 Add User Name to WSS Entry



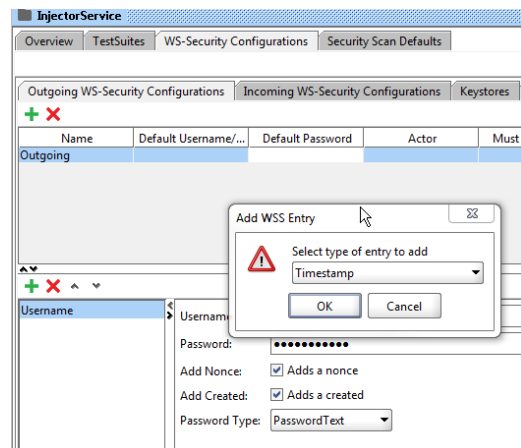
- f. Enter the Integration user's username and password for the integration user and set the Password Type to PasswordText. (The user was defined when deploying the Injector.)

Figure 6–11 Set Password Type



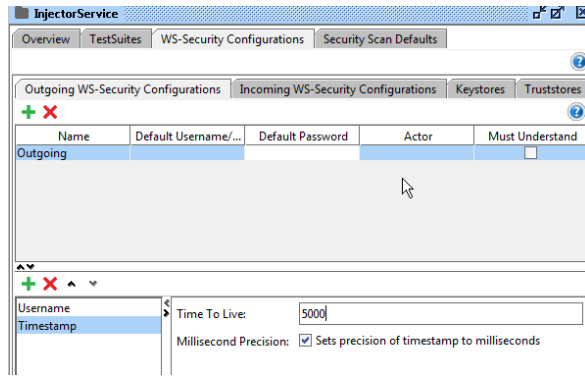
- g. Click the Plus sign in the lower section to create a timestamp WSS entry.

Figure 6–12 Create Timestamp WSS Entry



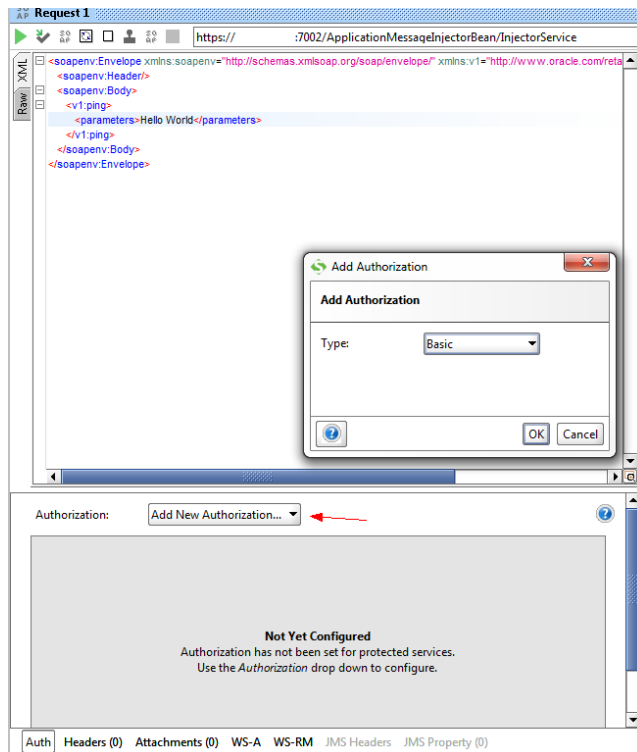
- h. Set the time to live to a large enough number to account for any network latency.

Figure 6–13 Set Time to Live Entry



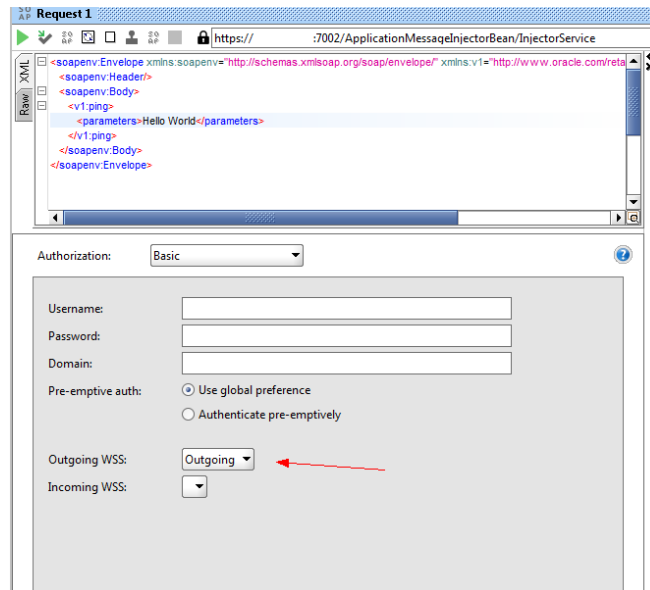
- i. The Inject Service has two operations. For each Operations' Request.
 - Add a New Authorization: Basic

Figure 6–14 Add New Authorization



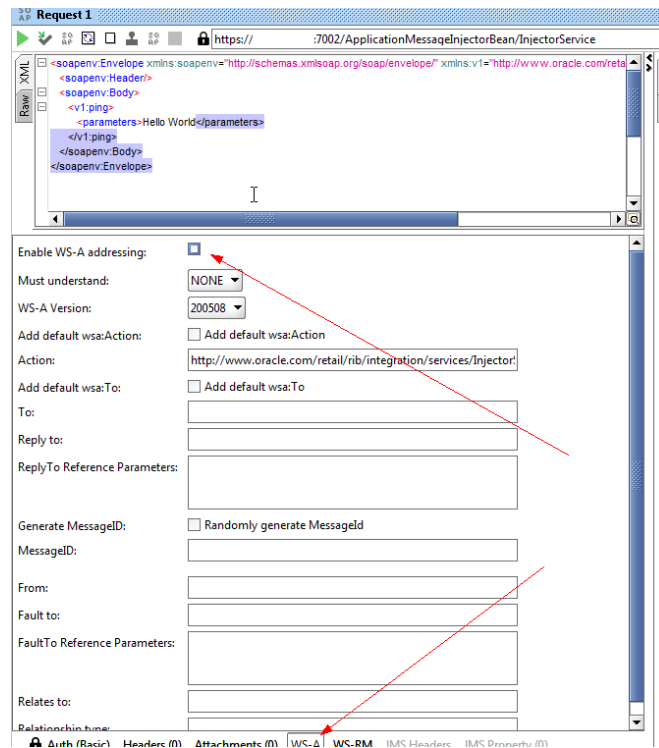
- Select the name you used for the Outgoing WSS.

Figure 6–15 Outgoing WSS



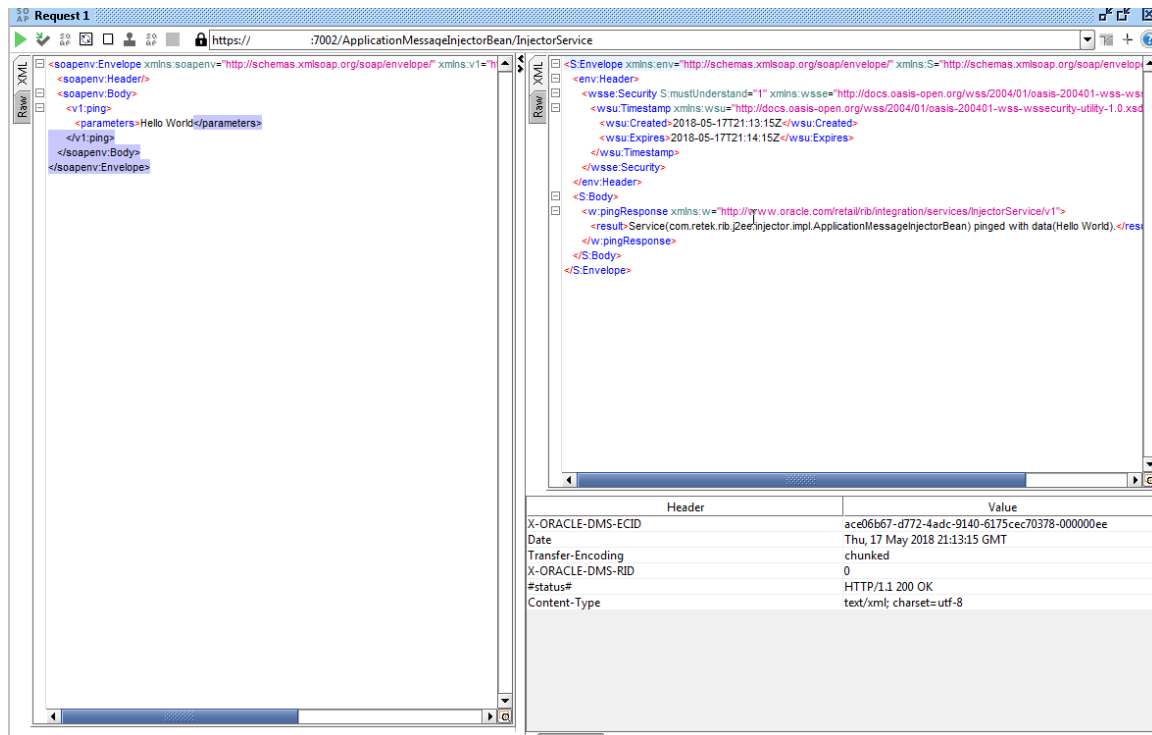
- j. On the WS-A tab make sure Enable WS-A addressing is not selected.

Figure 6–16 WS-A Tab



- k. Create a valid request and send it. The request is now using policy A.

Figure 6–17 Create Valid Request



OCDS (ORDS) Web Services

This chapter describes the process to deploy the configured 19.2 ords.war file onto the OCDS domain.

Prerequisites

The prerequisites and steps outlined in the OCDS Schemas chapter have been completed.

The target WebLogic Admin Server and Managed Server should be running.

The `JAVA_HOME` environment variable must be set.

Preparation

The OCDS Web Services leverage Oracle REST Data Services (ORDS). Perform the following procedure to prepare for the installation of ORDS.

1. Unzip `ocds-ords-deployment.zip`.
2. Copy the configured 19.2 ords.war file into the `/dist` folder.

Note: This ords.war file should have the `config dir` set to the correct `/config` folder. Otherwise set the location of the ORDS configuration files using:

```
java -jar ords.war configdir </path/to/ords/config>
```

3. Copy `/config` folder that was used when setting up ORDS in the database itself (see “Configured ORDS 19.2 for the OCDS database”). It should contain `/config/ords` that has `url-mapping.xml`, `<pdb_name>_pu.xml` and other config files created when setting up the database.

Deploy ORDS

Perform the following procedure to deploy the ORDS web application onto a WebLogic Domain:

1. Configure `conf/ords-deployment-env-info.json` file with the database and WebLogic domain details. This file is used by the deployment script.
 - a. Edit the Middleware Server definitions for `OrdsAppServer`.
 - `webLogicDomainName`: WebLogic domain name.

- webLogicDomainHome: WebLogic domain home directory.
- webLogicDomainAdminServerUrl: Server URL information.
- webLogicDomainAdminServerHost: Server host.
- webLogicDomainAdminServerPort: Admin Server port.
- webLogicDomainTargetManagedServerName: Managed Server name.

Figure 7–1 Middleware Server Definitions for OrdsAppServer

```
"OrdsDeploymentEnvInfo": {
  "MiddlewareServerDef": {
    "OrdsAppServer": {
      "weblogicDomainName": "ocds_domain",
      "weblogicDomainHome": "yu01/webadmin/products/wls_ocds/domains/ocds_domain",
      "weblogicDomainAdminServerUrl": "t3://localhost:8440",
      "weblogicDomainAdminServerProtocol": "t3",
      "weblogicDomainAdminServerHost": "localhost",
      "weblogicDomainAdminServerPort": "8440",
      "weblogicDomainAdminServerUserAlias": "OcdsAppServerAdminServerUserAlias",
      "weblogicDomainTargetManagedServerName": "OCDS_ManagedServer_1",

      "ordsIntegrationUserGroup": "OcdsMonitorGroup",
      "ordsIntegrationUserAlias": "IntegrationUserAlias",
      "ordsIntegrationUser": "GET_FROM_WALLET",
      "ordsIntegrationPassword": "GET_FROM_WALLET",
    }
  },
  "OrdsApplication": {
    "appName": "ords",
    "OrdsAppUses": [
      "OrdsAppServer"
    ]
  }
}
```

2. Stop and restart the Managed Server and the Admin Server.
3. With the WebLogic Admin Server and the Managed Server running, change to the ocds-ords-deployment/bin folder and execute the version ords-deployer script for the o/s using the switches:

```
-setup-credentials -deploy-ords-app
```

On Linux:

```
./ords-deployer.sh -setup-credentials -deploy-ords-app
```

On Windows:

```
ords-deployer.cmd -setup-credentials -deploy-ords-app
```

- a. There will be one prompt for WebLogic user credentials:
 - Enter username for alias (OcdsAppServerAdminServerUserAlias):
Enter the WebLogic Admin Server credentials.

Figure 7–2 WebLogic User Credentials

```
bash-4.2$ ./ords-deployer.sh -setup-credentials -deploy-ords-app
Extracting jars from jps-wallet-all.
log4j:WARN No appenders could be found for logger (com.oracle.retail.integration.common.security.credential.CredentialStoreManager).
log4j:WARN Please initialize the log4j system properly.

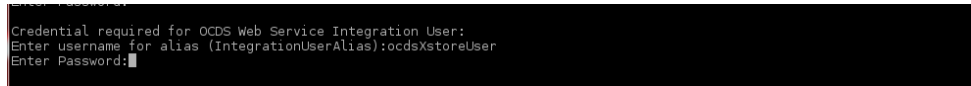
Credential required for weblogicDomainAdminServerHost(localhost) weblogicDomainAdminServerPort(8440):
Enter username for alias (OcdsAppServerAdminServerUserAlias):weblogic
Enter Password:
```

- b. There will be one prompt to create the OCDS Integration User:

Enter the credentials for the OCDS Integration user. These credentials will enable an Omnichannel application, such as the Xstore Suite, to communicate with OCDS.

Note: Password must not start with a number.

Figure 7–3 OCDS Integration User



Verify Installation

If the OCDS web services have been successfully installed then you should be able to request a JSON response from one of the OCDS REST resources.

1. Test by invoking a REST endpoint using a tool like curl (or SOAPUI, and so on). Curl is used for demonstration purposes.

URL

`http[s]://host[:port]/ords/<path-prefix>/omnichannel/metadata-catalog/`

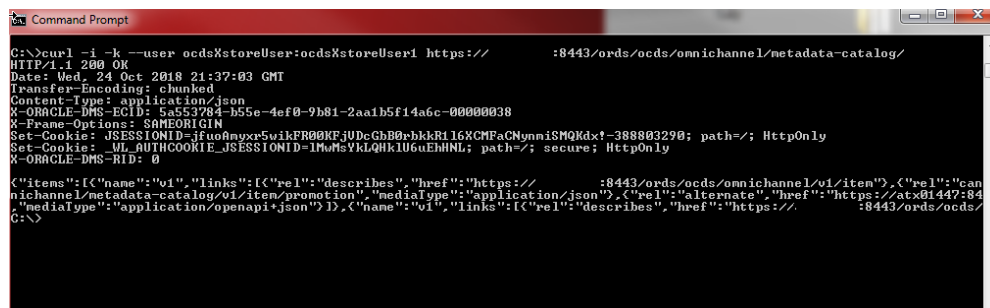
where

- `<path-prefix>` is the prefix (defined in a previous step) that must occur at the start of the request path

```
curl -i -k --user ocdsXstoreUser:ocdsXstoreUser1
```

```
https://example:8443/ords/ocds/omnichannel/metadata-catalog/
```

Figure 7–4 Request Path



Appendix A: Migrating OCDS to v19.1

Preliminary Requirements

To migrate OCDS using this document requires that the deployed version of OCDS is at least v16.0.045 HF3. If OCDS is not at this patch level, apply the hotfixes sequentially from the current hotfix level to the deployment until HF3 is reached. When the deployed OCDS is patched to a least v16 HotFix 3 level, follow the steps in this document.

Download the `OmnichannelCloudDataServices19.1.000ForAll19.x.xApps_eng_ga.zip`.

Extract the contents of the `OmnichannelCloudDataServices19.1.000ForAll19.x.xApps_eng_ga.zip` file.

The `<<Base Extract Dir>>` notation will be used in the documentation to provide the location of the extracted files.

There are three files in the extract which will be used for migrating the application:

`<<Base Extract Dir>>/ocds_home/ocds-db/ocds-database-creation.zip`

`<<Base Extract Dir>>/ocds_home/ocds-bdi/ocds-jobadmin-deployment.zip`

`<<Base Extract Dir>>/ocds_home/ocds-rib/ocds-injector-deployment.zip`

OCDS Schema Upgrade

This section describes the OCDS Schema Upgrade.

Pre Schema Upgrade

Shutdown the `ocds-jobAdmin` and `ocds-injector` servers before applying the update.

OCDS Schema Migration

Extract the contents of the `<<Base Extract Dir>>/ocds_home/ocds-db/ocds-database-creation.zip` file. The location of the database creation files extract is identified as `<<DB Extract Dir>>` in the schema migration steps.

Connect to the `ocds_ifc` schema and run the following scripts:

1. <<DB Extract Dir>>/scripts/ocds_txn/migration/drop-ocds-ifc-input-tables.sql. This will drop the ocds_ifc.*_IN tables in the ocds_ifc schema.
2. <<DB Extract Dir>>/scripts/rtg_ifc/ddl/bdi_ocds_ddl.sql. Provided by RTG to create the interface _IN tables in the ocds_ifc schema.

Connect to the ocds_ifc schema as a user able to grant privileges to the ocds_txn user.

1. <<DB Extract Dir>>/scripts/ocds_txn/plsql/Interface_Schema_Access.sql script. Grants permissions to the ocds_txn user to access data in the ocds_ifc tables.

Connect to the ocds_txn schema and run the following scripts:

1. <<DB Extract Dir>>/scripts/ocds_txn/ddl/ocds-txn-ddl-16-to-19.sql. Migrates the ocds_txn ddl from v16 hf03 to OCDS v19.1.
2. <<DB Extract Dir>>/scripts/ocds_txn/plsql/ocds-txn-plsql.sql. Creates the v19.1 pl sql packages used for import and utility operations.
3. <<DB Extract Dir>>/scripts/ocds_txn/rest/ocds-enable-rest.sql. Creates the 19.1 versions of the rest plsql packages.

Connect to the ocds_txn schema as a user able to grant privileges to the ocds_ifc user.

1. <<DB Extract Dir>>/scripts/ocds_ifc/plsql/Transaction_Schema_Access.sql. Grant the ocds_ifc user permission to access the ocds_txn importer packages.

Connect to the ocds_ifc schema and run the following script:

1. <<DB Extract Dir>>/scripts/ocds_ifc/plsql/ocds-ifc-plsql. Replaces the base rtg importers with the ocds ocds_ifc 19.1 importer packages.

Post Schema Upgrade

Connect to the ocds_txn schema and run the following script:

<<DB Extract Dir>>/scripts/ocds_txn/ddl/validate-ddl-16-to-19.sql. Validates that the expected database objects for the ocds_txn schema for 19.1 are present. Missing database objects will generate an error message.

Restart the jobAdmin and ocds-injector servers.

This completes the steps for migrating the ocds_txn and ocds_ifc schemas to the current release of OCDS v19.1.xxx.

WAR FILE Upgrades

This section describes the .war file upgrade.

OCDS jobAdmin.warUpgrade

Extract the contents for the <<Base Extract Dir>>/ocds_home/ocds-bdi/ocds-jobadmin-deployment.zip file.

The location of the extracted files is identified as <<BDI Extract Dir>>.

Edit Deployment Configuration File

1. Edit the <<BDI Extract Dir>>/conf/bdi-job-admin-deployment-env-info.json file with the database and WebLogic domain details. This file is used by the

deployment script to configure WebLogic. This information can be merged from a previous version of the `bdi-job-admin-deployment-env-info.json` used in previous deployment. For additional information about the `/bdi-job-admin-deployment-env-info.json` refer to the *Oracle Retail Omnichannel Cloud Data Service Installation Guide* version 19.1.

a. Edit the Datasource definitions for JobAdminDataSource:

- `jdbcUrl`: Configure the `jdbcUrl` for all DataSources definitions in `DataSourceDef`.

`BatchInfraDataSource` references a schema created using the WebLogic RCU (`<prefix>_WLS`).

All other datasources reference the OCDS interface (`ocds_ifc`) schema created during the prerequisite step: OCDS Database Creation.

Figure A–1 Datasource Definitions

```

1 "BdiJobAdminDeploymentEnvInfo": {
  "DataSourceDef": {
    "JobAdminDataSource": {
      "dataSourceName": "OcdsJobAdminDataSource",
      "dataSourceClass": "oracle.jdbc.pool.OracleDataSource",
      "dataSourceJndiName": "jdbc/OcdsJobAdminDataSource",
      "jdbcUrl": "jdbc:oracle:thin://:1521/ocdspdb",
      "jdbcUserAlias": "ocdsJobAdminDataSourceUserAlias",
      "jdbcUser": "GET_FROM_WALLET",
      "jdbcPassword": "GET_FROM_WALLET",
      "dataSourceProperties": {
        "connectionPool_MaxCapacity": "300"
      }
    },
    "ReceiverServiceDataSource": {
      "dataSourceName": "OcdsReceiverServiceDataSource",
      "dataSourceClass": "oracle.jdbc.pool.OracleDataSource",
      "dataSourceJndiName": "jdbc/OcdsReceiverServiceDataSource",
      "jdbcUrl": "jdbc:oracle:thin://:1521/ocdspdb",
      "jdbcUserAlias": "ocdsReceiverServiceDataSourceUserAlias",
      "jdbcUser": "GET_FROM_WALLET",
      "jdbcPassword": "GET_FROM_WALLET",
      "dataSourceProperties": {
        "connectionPool_MaxCapacity": "300"
      }
    },
    "BatchInfraDataSource": {
      "dataSourceName": "BatchInfraDataSource",
      "dataSourceClass": "oracle.jdbc.xa.client.OracleXADataSource",
      "dataSourceJndiName": "jdbc/BatchInfraDataSource",
      "jdbcUrl": "jdbc:oracle:thin://:1521/ocdspdb",
      "jdbcUserAlias": "batchInfraDataSourceUserAlias",
      "jdbcUser": "GET_FROM_WALLET",
      "jdbcPassword": "GET_FROM_WALLET",
      "dataSourceProperties": {
        "connectionPool_MaxCapacity": "300"
      }
    },
    "JobXmlDataSource": {
      "dataSourceName": "JobXmlDataSource",
      "dataSourceClass": "oracle.jdbc.xa.client.OracleXADataSource",
      "dataSourceJndiName": "jdbc/JobXmlDataSource",
      "jdbcUrl": "jdbc:oracle:thin://:1521/ocdspdb",
      "jdbcUserAlias": "jobXmlDataSourceUserAlias",
      "jdbcUser": "GET_FROM_WALLET",
      "jdbcPassword": "GET_FROM_WALLET",
      "dataSourceProperties": {
        "connectionPool_MaxCapacity": "300"
      }
    }
  }
},

```

b. Edit the Middleware Server definitions for JobAdminAppServer

- `webLogicDomainName`: WebLogic domain name.
- `webLogicDomainHome`: WebLogic domain home directory.
- `webLogicDomainAdminServerUrl`: Server URL information.
- `webLogicDomainAdminServerHost`: Server host.
- `webLogicDomainAdminServerPort`: Admin Server port.
- `webLogicDomainTargetManagedServerName`: Managed Server name.

- jobAdminUiUrl: Host and managed server port where Job Admin application will be deployed. This can be setup with the HTTPS port.

Figure A–2 OCDS Setup HTTPS Port

```

^ "MiddlewareServerDef":{
  "JobAdminAppServer": {
    "weblogicDomainName": "ocds_domain",
    "weblogicDomainHome": "jw01/webadmin/products/wls_ocds/domains/ocds_domain",
    "weblogicDomainAdminServerUrl": "t3://localhost:8440",
    "weblogicDomainAdminServerProtocol": "t3",
    "weblogicDomainAdminServerHost": "localhost",
    "weblogicDomainAdminServerPort": "8440",
    "weblogicDomainAdminServerUserAlias": "OcdsAppServerAdminServerUserAlias",
    "weblogicDomainTargetManagedServerName": "OCDS_ManagedServer_1",

    "jobAdminUiUrl": "http://localhost:8442/ocds-batch-job-admin",
    "jobAdminUiUserGroup": "BdiEdgeOcdsJobAdminGroup",
    "jobAdminUiUserAlias": "ocdsJobAdminUiUserAlias",
    "jobAdminUiUser": "GET_FROM_WALLET",
    "jobAdminUiPassword": "GET_FROM_WALLET",

    "jobOperatorUiUserGroup": "BdiEdgeOcdsJobOperatorGroup",
    "jobOperatorUiUserAlias": "ocdsJobOperatorUiUserAlias",
    "jobOperatorUiUser": "GET_FROM_WALLET",
    "jobOperatorUiPassword": "GET_FROM_WALLET",

    "jobMonitorUiUserGroup": "BdiEdgeOcdsJobMonitorGroup",
    "jobMonitorUiUserAlias": "ocdsJobMonitorUiUserAlias",
    "jobMonitorUiUser": "GET_FROM_WALLET",
    "jobMonitorUiPassword": "GET_FROM_WALLET"
  }
},
"JobAdminApplication": {
  "appName": "ocds",
  "JobAdminAppUses": {
    "JobAdminDataSource": {
      "JobAdminAppServer",
      {
        "RemoteJobAdminAppServers": []
      }
    }
  }
}

```

c. Edit RMS JobAdmin Server.

- jobAdminUiUrl: Host and managed server port where Job Admin application will be deployed. This can be setup with the HTTPS port.

Figure A–3 RMS JobAdmin Server Setup

```

"RmsJobAdminAppServer": {
  "jobAdminUiUrl": "http://localhost:7009/rms-batch-job-admin",
  "jobAdminUiUserAlias": "rmsJobAdminBaseUrUserAlias",
  "jobAdminUiUser": "GET_FROM_WALLET",
  "jobAdminUiPassword": "GET_FROM_WALLET",
}

```

Job Admin Deployment

Perform the following procedure to install and deploy the Job Admin Application.

1. Change to the <<BDI Extract Dir>>/ocds-jobadmin-deployment/bin folder and execute the version bdi-job-admin-deployer script for the o/s using the switches:

```
-use-existing-credentials -run-db-schema-migration -deploy-job-admin-app
```

On Linux:

```
./bdi-job-admin-deployer.sh -use-existing-credentials -run-db-schema-migration
-deploy-job-admin-app
```

2. Bounce the application server running the bdi-batch-job-admin-19.1.000.war file.

OCDS ocds-injector.war Upgrade

This section describes the OCDS ocds-injector.war upgrade

Preparation

Extract the contents of the ocds-injector-deployment.zip file. The destination directory will be designated as <<INJ Extract Dir>>.

Edit Deployment Configuration Files

Edit the <<INJ Extract Dir>>/conf/ injector-deployment-env-info.json file with the database and WebLogic domain details. This file is used by the deployment script. This information can be merged from a previous version of the injector-deployment-env-info.json used in previous deployment. For additional information about the injector-deployment-env-info.json refer to the *Oracle Retail Omnichannel Cloud Data Service Installation Guide v19.1*.

1. Edit the Datasource definitions for InjectorDataSource.
 - a. jdbcUrl: This is the jdbc URL needed to connect to the OCDS Transactional schema. The OCDS Transactional (ocds_txn) schema was created during the prerequisite step: OCDS Database: Database Creation.

Figure A-4 jdbc URL

```
"InjectorDeploymentEnvInfo": {
  "DataSourceDef": {
    "InjectorDataSource": {
      "dataSourceName": "InjectorDataSource",
      "dataSourceClass": "oracle.jdbc.pool.OracleDataSource",
      "dataSourceJndiName": "jdbc/InjectorDataSource",
      "jdbcUrl": "jdbc:oracle:thin:@//:1521/ocdspdb",
      "jdbcUserAlias": "InjectorDataSourceUserAlias",
      "jdbcUser": "GET_FROM_WALLET",
      "jdbcPassword": "GET_FROM_WALLET"
    }
  }
},
```

- b. Edit the Middleware Server definitions for InjectorAppServer.
 - webLogicDomainName: WebLogic domain name.
 - webLogicDomainHome: WebLogic domain home directory.
 - webLogicDomainAdminServerUrl: Server URL information.
 - webLogicDomainAdminServerHost: Server host.
 - webLogicDomainAdminServerPort: Admin Server port.
 - webLogicDomainTargetManagedServerName: Managed Server name.

Figure A-5 Middleware Server Definition

```
"MiddlewareServerDef": {
  "InjectorAppServer": {
    "webLogicDomainName": "ocds_domain",
    "webLogicDomainHome": "/u01/...",
    "webLogicDomainAdminServerUrl": "t3://localhost:8440",
    "webLogicDomainAdminServerProtocol": "t3",
    "webLogicDomainAdminServerHost": "localhost",
    "webLogicDomainAdminServerPort": "8440",
    "webLogicDomainAdminServerUserAlias": "OcdsAppServerAdminServerUserAlias",
    "webLogicDomainTargetManagedServerName": "OCDS_ManagedServer_1",

    "injectorIntegrationUserGroup": "IntegrationGroup",
    "injectorIntegrationUserAlias": "IntegrationUserAlias",
    "injectorIntegrationUser": "GET_FROM_WALLET",
    "injectorIntegrationPassword": "GET_FROM_WALLET",
  }
},
```

OCDS Injector Deployment

Perform the following procedures to install and deploy the Injector application.

1. Change to the <<INJ Extract Dir>>/ocds-injector-deployment/bin folder and execute the version of injector-deployer script for the o/s using the switches:

```
-use-existing-credentials -deploy-injector-app
```

On Linux:

```
./injector-deployer.sh -use-existing-credentials -deploy-injector-app
```

2. Bounce the application server running the ocds-injector.war file.

ORDS Upgrade

The supported version of Oracle of ORDS is 19.4. To determine what version of ords is currently deployed, run the following from the command line:

```
java -jar ords.war version
```

If the deployed ords.war file is less than 19.4, download ords 19.4 from

<https://www.oracle.com/database/technologies/appdev/rest-data-services-1946-downloads.html>.

Follow the instructions provided in the ORDS documentation for performing the upgrade. Bounce the managed server running the ords.war file after the upgrade.