Oracle® Cloud

Getting Started with Remote Data Connector for Oracle® Business Intelligence Cloud Service

E67875-05

February 2017

To enable access to remote data sources in an on-premises network from Oracle BI Cloud Service, you must deploy and configure Oracle BI Cloud Service Remote Data Connector (Remote Data Connector) in your on-premises network for secure access to your data. This document introduces you to Remote Data Connector and provides instructions for downloading, deploying, and configuring it to access on-premises data sources.

Audience

The intended audience for these instructions is administrators who want to set up Oracle Business Intelligence Cloud Service Remote Data Connector to enable secure access from the cloud to on-premises relational data sources for analysis.

About Oracle Business Intelligence Cloud Service Remote Data Connector

Oracle Business Intelligence Cloud Service Remote Data Connector (Remote Data Connector) enables secure connection to on-premises data sources for analysis in the cloud.

Remote Data Connector works with the BI Server Data Gateway running in the Oracle BI Cloud Service environment to provide secure access to on-premises data using private/public key pairs and SSL communication.

Supported Data Sources

Remote Data Connector supports these on-premises database types:

- DB2
- Oracle
- Oracle OLAP

- SQL Server
- Teradata

Supported Web Servers

In addition to Oracle Weblogic Server, RDC supports Apache Tomcat (for configuration details, see Setting Up RDC With Apache Tomcat).

About SSL Requirements

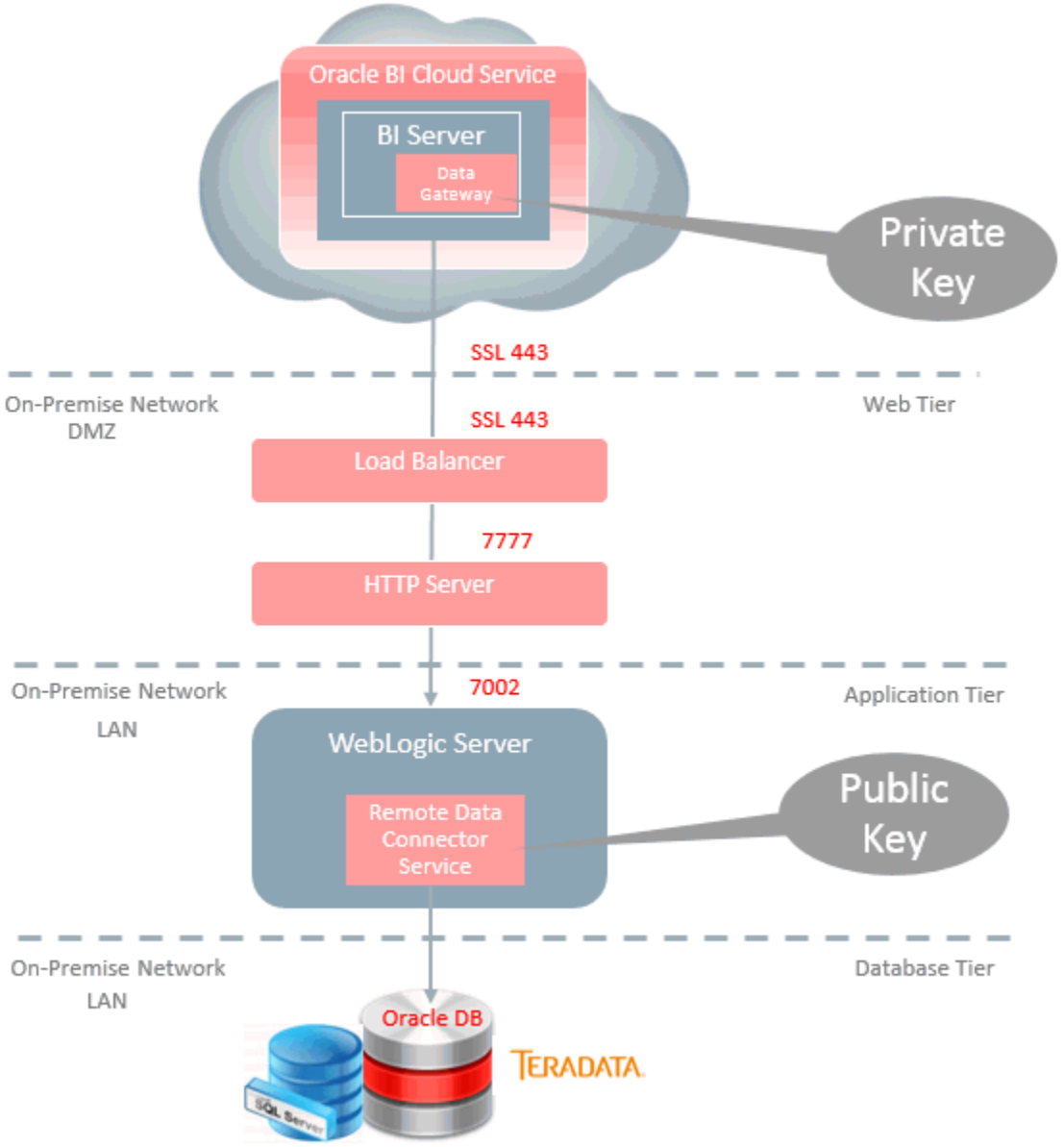
For detailed information about SSL requirements for RDC, refer to the following knowledge article on My Oracle Support:

BICS RDC: SSL Requirements for BI Cloud Service Remote Data Connector (Doc ID 2227789.1)

Architecture

Each Oracle BI Cloud Service instance is provisioned with a unique private key. A public key is available for download from Oracle BI Cloud Service Console. This public key when deployed on-premises in Remote Data Connector enables Remote Data Connector to verify the authenticity of a query received from a BI Server in Oracle BI Cloud service. SSL configured on-premises at a Load Balancer or HTTP servers provides secure access to on-premises data.

This diagram shows a typical on-premises network architecture. Contact your network administrator for additional details about your network configuration.



Prerequisites

Before you set up Oracle BI Cloud Service Remote Data Connector, make sure that your environment meets these requirements.

- If you are deploying RDC using Oracle Weblogic Server, then download and install Weblogic Server from Oracle Technology Network (OTN). You must also deploy Node Manager with Weblogic Server. Alternatively, install Apache Webcat (for configuration details, see Setting Up RDC With Apache Tomcat).
- Obtain the public IP and domain name.

- Download the Oracle BI Cloud Service Remote Data Connector WAR file (obi-remotedataconnector.war) from OTN.
- Configure your load balancer or HTTP server for SSL communication.
- Download and install Oracle Business Intelligence Developer Client Tool (12.2.1.0.0) from OTN.

Configure Secure Access to On-Premises Data

This topic describes the steps required to deploy Oracle BI Cloud Service Remote Data Connection (RDC).

You install RDC on an application server, set up and test connections to the data source that you want to query for analysis, and update the BI Metadata Repository to include the new connection information in the appropriate connection pool.

To configure secure access to on-premises data:

1. Deploy the Remote Data Connector WAR file.
    a. Log in to Oracle WebLogic Server and, in the Change Center pane, click **Lock & Edit**.
    b. In the Domain Structure, select **Deployments**.
    c. In the Deployments list, click **Install**.
    d. In the Install Application Assistant, click the **upload your file(s)** link, click the **Choose File** button for the Deployment Archive, and select the `obi-remotedataconnector.war` file you downloaded.
    e. Click **Next** twice.
    f. Confirm that the **Install this deployment as an application** option is selected and click **Next**.
    g. Select the appropriate server target.
    h. Click **Next**.
    i. Verify the deployment summary.
    j. Click **Finish**.

       You see a message that indicates that the deployment has been successfully installed.
    k. On the left-hand side Change Center pane, click **Activate Changes**.
    l. On the right-hand content pane, select the option next to the EAR just deployed.
    m. Click **Start** to view the drop-down list and select **Servicing all requests**.
    n. In the content pane of the new page, click **Yes**.
    o. Disable temporarily Remote Data Connector's Metadata security:

If Node Manager is installed, then in Oracle WebLogic Console navigate to Environment (on the left pane), then Servers. Select the server to which Remote Data Connector was deployed, display the Server Start tab, then in the Arguments edit box, add:

```
-Doracle.bics.rdc.disable_metadata_security=1
```

Note: If this variable is not set to 1, the status URL (provided in the next step) is blocked. After setting this property, you must restart Oracle WebLogic Server. When Oracle WebLogic Server is started subsequently, it has to be done using Node Manager.

If Node Manager is not installed, then before starting Oracle WebLogic Server (in the same command prompt or script that starts Oracle WebLogic Server), execute one of the following commands to set this environment variable:

On Linux:

```
export DISABLE_RDC_METADATA_SECURITY=1
```

On Windows:

```
set DISABLE_RDC_METADATA_SECURITY=1
```

p. Test the deployment by navigating to `http(s)://<weblogic-server>:<weblogic-port>/obiee/javads?status`.

You should see an XML file. If you see an error "401—Unauthorized", then verify step p and retry.

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<JavaDSServer>
  ▼<Services>
     <Service name="oracle.bi.datasource.service.DatasourceService"
     processor="oracle.bi.datasource.service.DatasourceServiceProcessor"/>
   </Services>
  ▼<Cartridges>
   ▼<Cartridge name="JDBC" uuid="fd35144d-26f1-491e-936e-17039604fec6" version="12.1">
      <Connector name="JDBC (Direct Driver)" uuid="5e9ffb28-b5ce-4201-b1a6-8f9f585389ea" version="12.1"/>
      <Connector name="JDBC (JNDI)" uuid="b41b07f5-7c55-4adf-8c51-ebe9a09b37f7" version="12.1"/>
    </Cartridge>
   </Cartridges>
  ▼<ConfigSources>
   ▼<ServiceProperties>
     ▼<![CDATA[
        #DatasrcServer/src/jdbc-only-serviceprocessor.properties
        oracle.bi.datasource.service.DatasourceService=oracle.bi.datasource.service.DatasourceServiceProcessor
      ]]>
    </ServiceProperties>
   ▼<CartridgeProperties>
     ▼<![CDATA[
        #DatasrcServer/src/jdbc-only-cartridges.properties jdbc=oracle.bi.datasource.jdbc.JDBCCartridge
      ]]>
    </CartridgeProperties>
   </ConfigSources>
</JavaDSServer>
```


2. Add the JDBC data source.

   a. Log in to Oracle WebLogic Server and, in the Domain Structure, select **Services**.

b. In the Summary of Services list, click **Data Sources**.

c. In the Configuration tab, under Data Sources, click **New** and select the **Generic Data Source** option.

d. Enter the Name and JNDI Name fields and click **Next**. To avoid confusion, use the same name. The JNDI Name forms a component of the URL used to access this data source after the setup is complete.

Note: Make a note of the name you enter to reuse later when you set the URL for the remote data connection. For example, you might use mysalesdatasource as a name for your sales database.

---

e. In the next screen, for an Oracle database, select **Oracle's Driver (Thin) for Service connections: Versions:Any** in the Database Driver drop-down list and click **Next**.

f. Accept defaults in the next screen and click **Next**.

g. Enter your database connection details in the next screen and click **Next**.

h. In the next screen, click **Test Configuration** to test your database connection.

i. Once you receive the Connection test succeeded message, click **Next**.

j. In the Targets tab under the settings, select appropriate target server for the JDBC Data source.

k. Verify that you can see the newly created JDBC Data source in the list of Data Sources.

3. Download and deploy the public key.

   a. Log in to Oracle BI Cloud Service.

   b. Navigate to the Oracle BI Cloud Service Console.

   c. Click **Connections.**

   d. Click **Get Public Key** to download the public key.

   e. When downloading the public key, in the Save dialog, make that sure the name is oracle_bics_rdc.pem and save it to your local machine.

   f. Copy `oracle_bics_rdc.pem` to the Oracle WebLogic Server environment in the DOMAIN_HOME/rdc_keys/<deployment_name> folder (by default the deployment name is “obi-remotedataconnector”). The folder “rdc_keys/<deployment_name>” is created by RDC the first time it is deployed. The `DOMAIN_HOME` path is the directory in which the Oracle WebLogic Server domain is installed.

4. Make Remote Data Connector available to Oracle BI Cloud Service with the help of a network administrator.

a. Configure the load balancer/reverse proxy for SSL communication and to route requests to the HTTP Server.

b. Configure the HTTP server to direct requests to Oracle WebLogic Server.

c. Test Remote Data Connector with a public IP address/domain URL, for example `https://<Public IP or Domain Name>:<port>/obiee/javads?status`.

   Note: If Remote Data Connector metadata security is not disabled, then this URL fails with the message 401-Unauthorized. To disable Remote Data Connector metadata security, set `oracle.bics.rdc.disable_metadata_security` to 1 or set the `DISABLE_RDC_METADATA_SECURITY` environment variable to 1.

5. Update the data model file connection pool.

   a. Open the Oracle Business Intelligence Developer Client Tool.

   b. In the **File** menu, select **Load Java Datasources** to obtain the properties of Remote Data Connector.

   c. In the Connect to Java Datasource Server dialog, enter the public IP address or domain name for the Hostname and the Port where Remote Data Connector is running. For **Username**, enter "rdcuser", and for **Password**, enter "rdcpwd".

   d. Click **OK**.

      A message is displayed indicating, "Successfully loaded javads metadata from `https://<Public IP or Domain Name>:<Port>`". If Remote Data Connector metadata security is not disabled, then this URL fails with the message 401-Unauthorized. To disable Remote Data Connector metadata security, set `oracle.bics.rdc.disable_metadata_security` to 1 or set the `DISABLE_RDC_METADATA_SECURITY` environment variable to 1.

   e. Open the data model file BI Metadata Repository in Offline mode.

   f. In the Physical layer, edit the connection pool.

   g. In the Connection Pool dialog, change the call interface to **JDBC (JNDI)**.

   h. Change the Data source name to the Remote Data Connector URL.

      The Remote Data Connector URL is the endpoint URL that you created earlier. You must specify the URL in the format: `https://<Public IP or Domain Name>:<port>/obiee/javads/<JNDI connectionname>`. Note that myjdbcdatasource was specified in Step 2.

   i. Switch to the **Miscellaneous** tab.

      This step is mandatory because the BI Metadata Repository is not be updated correctly unless the **Use SQL over HTTP** variable is set to `true`. This is only saved on switching to this tab.

   j. Click **OK**.

k. Save the data model file.

6. Test that the URL entered in the **Data source name** of the Connection Pool is accessible through the internet.

   Tip: An easy way to test this is to use a smart phone with its mobile data/3G/4G turned on and WiFi turned off, then trying to go to the exact URL entered in the **Data source name** field in step 5 (i). If everything is working correctly, then the message "401--Unauthorized" should be displayed. For other errors, please double check the protocol (HTTP/HTTPS), URL and port.

7. Upload the data model file to Oracle BI Cloud Service. For information about uploading the data model, see 'About Uploading Data Models to the Cloud' in the *Using Oracle Business Intelligence Cloud Service* guide.

You have now configured secure access to relational data sources.

Setting Up RDC With Apache Tomcat

To deploy RDC with the Apache Tomcat web server, first install and configure Apache Tomcat as described in Apache Tomcat documentation. Then complete the setup by following the steps in this section to configure the Data Source Name (DSN), BI Metadata Repository, and the RDC metadata security.

Prequisites

Before you start, install:

- Apache Tomcat for Windows or Linux Version 7.0.x or later to a maximum of Version 9.0.0.x.
- Oracle BI Administration Tool Version 12.2.1.0.0.

  Download Oracle BI Administration Tool using the **Oracle Business Intelligence Developer Client Tool (12.2.1.0.0)** download link at http://www.oracle.com/technetwork/middleware/bi-enterprise-edition/downloads/business-intelligence-2717951.html.

Steps For Setting Up RDC With Apache Tomcat

1. Configure the Data Source Name, see Configuring the Data Source Name.
2. Configure metadata security, see Configuring Metadata Security.
3. Configure how Apache Tomcat is deployed, see Deploying the RDC Application.
4. Configure the BI Metadata Repository, see Configuring the BI Metadata Repository.

Configuring the Data Source Name (DSN)

When you set up Apache Tomcat, configure the DSN to match the DSN that is specified in the BI Metadata Repository Connection Pool.

1. Create the following Resource under the GlobalNamingResources element (if necessary, create a GlobalNamingResources element), by editing <*Tomcat_installation_directory*>/conf/server.xml as follows:

```
<GlobalNamingResources>
<!-- Editable user database that can also be used by
UserDatabaseRealm to authenticate users
-->
<Resource
name="jdbc/myoracle"
global="jdbc/myoracle"
auth="Container"
type="javax.sql.DataSource"
driverClassName="oracle.jdbc.OracleDriver"
url="jdbc:oracle:thin:@biserver-dev.us.oracle.com:1521:orcl"
username="northwind"
password="n"
maxActive="15"
maxIdle="1"
maxWait="-1" />
</GlobalNamingResources>
```

2. **Specify the context, by editing <*Tomcat_installation_directory*>/conf/context.xml as follows:**

```
<Context>
<ResourceLink name="myjdbcdatasource"
global="jdbc/myoracle"
auth="Container"
type="javax.sql.DataSource" />
<!-- Default set of monitored resources -->
<WatchedResource>WEB-INF/web.xml</WatchedResource>
</Context>
```

Configuring Metadata Security

When you set up Apache Tomcat, you must disable metadata security.

1. Disable metadata security in one of the following ways:

   - By executing a command in the context of the shell that launches Apache Tomcat.

     Set the environment variable DISABLE_RDC_METADATA_SECURITY in the context of the shell that launches Tomcat. For example:

     (On Linux) `export DISABLE_RDC_METADATA_SECURITY=1`

     (Windows) `set DISABLE_RDC_METADATA_SECURITY=1`

   - By setting the Java property `oracle.bics.rdc.disable_metadata_security` in the catalina.sh/bat file, as shown in the following snippet from the catalina.sh/bat file:

     ```
     if [ -z "$LOGGING_MANAGER" ]; then JAVA_OPTS="$JAVA_OPTS -
     Djava.util.logging.manager=org.apache.juli.ClassLoaderLogMa
     nager" else JAVA_OPTS="$JAVA_OPTS $LOGGING_MANAGER" fi
     JAVA_OPTS="$JAVA_OPTS -
     Doracle.bics.rdc.disable_metadata_security=1"
     ```

     Note: The final line in the snippet disables metadata security to display the RDC application status page and enable connection from Oracle BI Administration Tool to Load Java Datasources.

Deploying the RDC Application

When you set up Apache Tomcat, you must deploy the RDC Tomcat application WAR file `obi-remotedataconnector-tomcat.war` with the context path:

```
/obiee
```

For example, your RDC application's root folder might be:

```
$MY_HOME/webapps/obiee
```

Configuring the BI Metadata Repository

When you set up Apache Tomcat, you configure the BI Metadata Repository and configure a public key.

1. Configure the BI Metadata Repository as follows:
   a. In Oracle BI Administration Tool, navigate to **File**, then **Load Java Datasources**.

      If you do not see the **Load Java Datasources** option, upgrade your Oracle BI Administration Tool to version 12.2.1.0.0 from OTN (http://www.oracle.com/technetwork/middleware/bi-enterprise-edition/downloads/business-intelligence-2717951.html).

   b. In the Connect to Java Datasource Server dialog, specify the hostname, port and username/password of the Tomcat server (including https if you are using a secure transfer protocol).

   c. When the Java data sources have loaded, open the BI Metadata Repository.

   d. Navigate to the **Connection Pool** area of the connection that you would like to use remotely.

      In the Connection Pool dialog, you will see two new Call Interfaces.

   e. Select **JDBC (JNDI)**.

   f. For the Data Source Name, specify:

      ```
      http(s)://<server-name>:<port>/obiee/javads/<JNDI
      connection name>
      ```

      For the DSN, specify:

      ```
      myjdbcdatasource
      ```

   g. On the Miscellaneous tab, make sure that the **SQL over HTTP** value is set to `true`.

   h. Save the metadata repository and deploy it in the BIServer/BICS instance/domain.

2. Download and deploy the public key.
   a. Log in to Oracle BI Cloud Service.

   b. Navigate to the Oracle BI Cloud Service Console.

c. Click **Connections.**

d. Click **Get Public Key** to download the public key.

e. When downloading the public key, in the Save dialog, make that sure the name is oracle_bics_rdc.pem and save it to your local machine.

f. Copy oracle_bics_rdc.pem to the Apache Tomcat environment in the folder `$CATALINA_HOME\webapps\obiee\rdc_keys`.

   The folder `obiee\rdc_keys` is created by RDC the first time it is deployed. If this path does not exist, verify whether RDC was deployed with the correct context path. The CATALINA_HOME path is the directory in which the Apache Tomcat is installed.

Oracle® Cloud Getting Started with Remote Data Connector for Oracle® Business Intelligence Cloud Service,
E67875-05