Securing Automic Automation Database Connections

Connecting to Oracle 12c Database

Version 1.1

Contents

Chapter 1: Introduction 4

Chapter 2: Configure TLS/SSL for the Oracle Database Server 5

2.1 Create and import the CA-signed certificate for the Database server 5

2.2 Configure the Oracle Server to use TLS/SSL 6

2.3 Configure the Oracle Client to connect to the Server via TLS/SSL 7

Chapter 3: Configure TLS/SSL for Automic 8

Chapter 1: Introduction

Oracle is one of the databases supported by Automic Automation for classical installations and Kubernetes clusters deployments. Many Cloud providers offer managed database services on platforms like Azure, Google Cloud Platform, and AWS and ensuring a secure connection to the database is no longer optional.

This document does not replace the Automic documentation or a basic understanding of the TLS/SSL protocol and other relevant concepts, such as private/public keys and certificates.

The following is only an example of configuring the Database Server and Automic to secure connections between the Automation Engine and an Oracle 12c database. Other setups might be a better fit based on your needs.

.

Chapter 2: Configure TLS/SSL for the Oracle Database Server

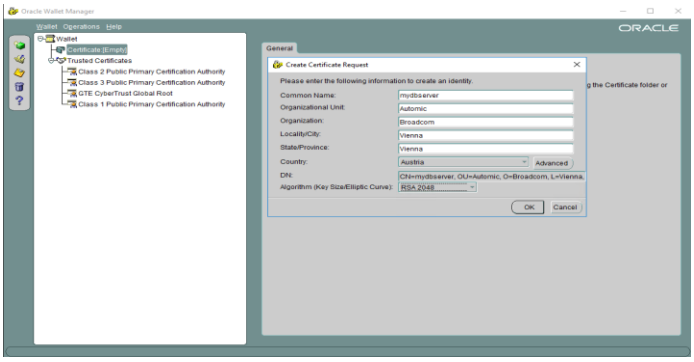
This guide uses a server certificate signed by an internal Certificate Authority (CA) to secure the connection between the database and Automation Engine.

Still, it is also possible to configure certificates that are self-signed or signed by a public CA. The Oracle documentation provides a detailed guide to configuring TLS/SSL connections.

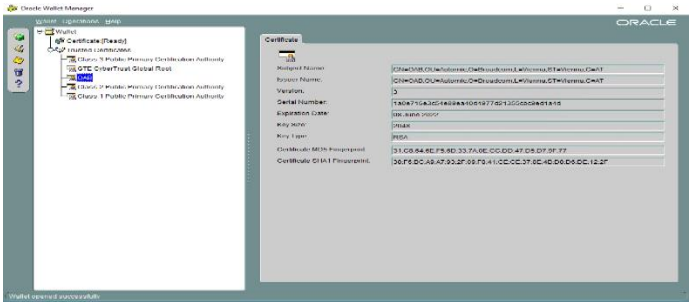
2.1 Create and import the CA-signed certificate for the Database server

Oracle uses wallets to store private keys and certificates that are required to secure communication between server and clients.

A wallet and a Certificate Signing Request (CSR) for a server certificate can be created with Oracle Wallet Manager as shown here.



The CSR can be exported and the certificate signed by the internal CA called OAB in the example below. The root (and intermediate if applicable) certificates of the CA that signed the server certificate need to be added to the Wallet truststore. Once the signed server certificate is available, it must be imported into the Wallet as well.



If hostname verification must be used, the Common Name (CN) in the server certificate must match the hostname/domain/address that the Automic system will use to connect to the Database Server. This can be achieved by specifying the server's distinguished name (DN) and TCPS as protocol in the client network configuration files.

In case of high availability servers with multiple hostnames or domain names, they can be included as Subject Alternative Names (SANs) during the creation of the certificate.

2.2 Configure the Oracle Server to use TLS/SSL

In order to configure the Database Server to use the previous Wallet, it is possible to either use Oracle Net Manager or directly edit the listener.ora and sqlnet.ora configuration files.

It is also possible to configure a list of cipher suites that can be used to secure the communication between the server and the client. More detailed information can be found in the official Oracle documentation.

In this guide, only server authentication is configured to secure connections and there are no restrictions when it comes to the encryption algorithms to be used.

The Database Server listener must include the path to the Wallet containing the keys/certificates and the TLS/SSL port to be used for TLS connections.

listener.ora:

```
SSL_CLIENT_AUTHENTICATION = FALSE

WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = C:\AWA\Main\db_certs)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = mydbserver)(PORT = 1521))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCPS)(HOST = mydbserver)(PORT = 2484))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC2484))
    )
  )
```

sqlnet.ora:

```
…
SQLNET.AUTHENTICATION_SERVICES= (BEQ, TCPS)

SSL_CLIENT_AUTHENTICATION = FALSE

WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
```

```
    (DIRECTORY = C:\AWA\Main\db_certs)
      )
  )
```

2.3 Configure the Oracle Client to connect to the Server via TLS/SSL

Since client authentication is not configured for the examples in this guide, it can be disabled in the sqlnet.ora file. To secure connections between the client and the server, a new entry must be included in tnsnames.ora to configure the TLS/SSL port and server DN if hostname verification should be used.

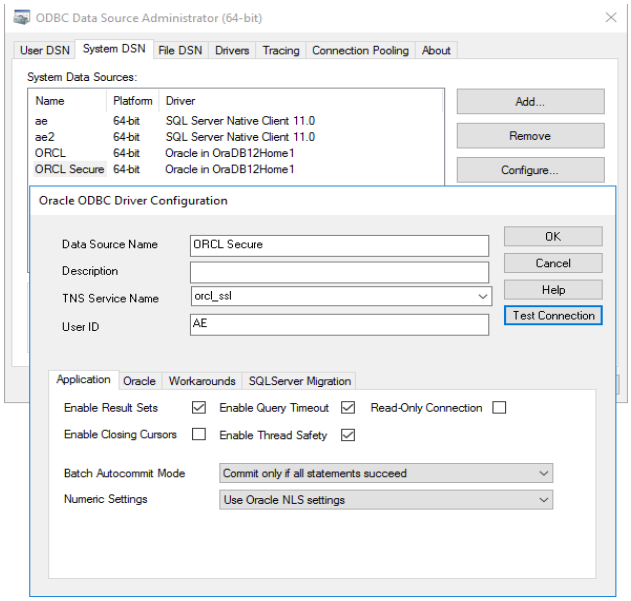
```
tnsnames.ora
…
ORCL_SSL =
 (DESCRIPTION =
  (ADDRESS_LIST =
   (ADDRESS = (PROTOCOL = TCPS)(HOST = mydbserver)(PORT = 2484))
        )
  (CONNECT_DATA =
   (SERVER = DEDICATED)
   (SERVICE_NAME = orcl.sbb01.spoc.global)
         (SECURITY=
   (SSL_SERVER_CERT_DN="cn=mydbserver,c=AT,o=Broadcom"))
  )
 )
```

In order to check that clients can connect to the database via TLS/SSL, you can run SQLPlus or the ODBC driver with the defined SSL connection.

The connection can be tested with SQLPlus from the command line as below:

```
> C:\Users\oab>sqlplus /@orcl_ssl
```

In order to use the ODBC Driver with the TLS/SSL option, a new data source can be included and the orcl_ssl TNS service name needs to be configured. The configured data source is then also used by the Automation Engine WPs and CPs to establish the connection to the database.



Chapter 3: Configure TLS/SSL for Automic

The Data source in the ucsrv.ini needs to be set to securely connect the WPs and CPs to use the previously configured ODBC data source:

ucsrv.ini:

```
[ODBC]
sqlDriverConnect=ODBCVAR=NNJNIORO,DSN=ORCL_SSL;UID=AE;PWD=oab;SP=NLS_LANGUAGE
=AMERICAN,NLS_TERRITORY=AMERICA,CODESET=WE8ISO8859P15
```

The Java server processes (JWPs and JCPs) connect to the Oracle Database Server via a JDBC driver that works with a JKS or PKCS12 Keystore or directly with the Oracle Wallet. The certificates required for the TLS/SSL authentication are stored in the corresponding Java Keystore.

This Oracle blog provides details on using the JDBC driver with different Keystores.
.
The Java Keystore in PKCS12 format can be created using Keytool Explorer. In this guide, only the CA root certificate used to sign the Database Server certificate are imported to the truststore. In the case of self-signed certificates, the clients use the server certificate for authentication.

If client-side authentication is required, besides the truststore with the necessary certificates for server authentication, a Keystore containing the client private key and certificate must be created and configured. This scenario is not part of this guide.

To enable TLS/SSL for Java processes, the JDBC DB connection string in the ucsrv.ini file has to be configured as below:

ucsrv.ini:

```
[JDBC]
sqlDriverConnect=jdbc:oracle:thin:@(description=(address=(protocol=tcps)(host=mydbserver)(port=2484)
)(connect_data=(sid=orcl)(SECURITY=(SSL_SERVER_CERT_DN="cn=mydbserver,c=AT,o=Broadcom")
)))
```

Lastly, the Java processes must be started with the following parameters to set the truststore containing the CA root certificate:

```
java -Xrs -Xmx1024M -
Djavax.net.ssl.trustStore=C:\AWA\Main\db_certs\db_client\dbserver_truststore.p12 -
Djavax.net.ssl.trustStoreType=PKCS12 -Djavax.net.ssl.trustStorePassword=changeit
-jar ucsrvjp.jar -IC:\AWA\Main\ServiceManager\bin\..\..\AutomationEngine\bin\ucsrv.ini -svc%port%
```

If the connections were successfully done via TLS/SSL, there should be messages similar to the ones below in the listener log (C:\app\botoa01\product\12.1.0\dbhome_1\log\diag\tnslsnr\mydbserver\listener\trace):

```
20-APR-2022 09:33:17 *
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=orcl.sbb01.spoc.global)(SECURITY=(S
SL_SERVER_CERT_DN=cn=mydbserver,c=AT,o=Broadcom))(CID=(PROGRAM=C:\AWA\Main\Automat
ionEngine\bin\UCsrvwp.exe)(HOST=mydbserver)(USER=SYSTEM))) *
(ADDRESS=(PROTOCOL=tcps)(HOST=fe80::e41a:c8d5:11:94f2%3)(PORT=55075)) * establish *
orcl.sbb01.spoc.global * 0
20-APR-2022 09:33:19 * service_update * orcl * 0

20-APR-2022 09:33:24 *
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=orcl.sbb01.spoc.global)(SECURITY=(S
SL_SERVER_CERT_DN=cn=mydbserver,c=AT,o=Broadcom))(CID=(PROGRAM=C:\AWA\Main\Automat
```

ionEngine\bin\UCsrvcp.exe)(HOST=mydbserver)(USER=SYSTEM))) *
(ADDRESS=(PROTOCOL=tcps)(HOST=fe80::e41a:c8d5:11:94f2%3)(PORT=55079)) * establish *
orcl.sbb01.spoc.global * 0
20-APR-2022 09:33:25 * service_update * orcl * 0

20-APR-2022 10:26:41 * (CONNECT_DATA=(CID=(PROGRAM=Automation Engine
JWP)(HOST=__jdbc__)(USER=mydbserver$))(sid=orcl)(SECURITY=(SSL_SERVER_CERT_DN=cn=my
dbserver,c=AT,o=Broadcom))) * (ADDRESS=(PROTOCOL=tcps)(HOST=10.49.164.113)(PORT=55899))
* establish * orcl * 0
20-APR-2022 10:27:22 * service_update * orcl * 0


BROADCOM
SOFTWARE