

Connecting to Oracle Database Using TNSNAMES.ORA in Java

Knock knock

by Przemysław Kruglej
11-2013

przemyslawkruglej.com
przemyslaw.kruglej@gmail.com

Table of Contents

<u>1 Utilizing TNSNAMES.ORA in Java.....</u>	<u>3</u>
<u>1.1 Knocking on Oracle's door.....</u>	<u>3</u>
<u>1.2 Meet TNSNAMES.ORA.....</u>	<u>3</u>
<u>1.3 The WHAT-WHAT (TNSNAMES.ORA) in Action.....</u>	<u>4</u>
<u>2 Further Reading & Useful Links.....</u>	<u>5</u>

1 Utilizing TNSNAMES.ORA in Java

- You have to use TNSNAMES.ORA.
- TNS-what?
- TNSNAMES.
- What-names?
- T N S N A M E S !
- What-what?

In the following thread on StackOverflow:

<http://stackoverflow.com/questions/19536162/how-to-connect-jdbc-to-tns-oracle>

a question concerning usage of *net service names*, stored in the TNSNAMES.ORA file, was asked. Let me briefly explain how to connect to Oracle using this approach.

1.1 Knocking on Oracle's door

While using JDBC, you have a few possibilities of establishing a connection. A common approach is to use the THIN driver which connects directly to Oracle using Java's sockets. To tell JDBC where you want to connect to, you provide a database URL of the following syntax:

```
jdbc:oracle:thin:@HOST:PORT:SID
```

where:

- jdbc:oracle:thin – defines that we want the JDBC Thin Driver to be used to establish the connection,
- HOST – address of the database server or the host's name,
- PORT – port on which the database server is listening for incoming connections,
- SID – Oracle System Identifier – identifies an instance of Oracle database.

However, instead of specifying HOST, PORT and SID values, we could provide the *net service name* from the TNSNAMES.ORA file. Let me just shortly explain what this fella' is all about!

1.2 Meet TNSNAMES.ORA

Put simply, TNSNAMES.ORA is a configuration file holding *net service names* which are aliases for database network addresses. Yep, *simply*. Each of the addresses is described by a connect descriptor composed of host address, port and SID or service name.

Those aliases are very useful – consider a situation, when the machine on which a database server is hosted changes – if you use an alias from TNSNAMES.ORA, you only have to change address of the database server there. On the other hand, if you would hardcode the host address in your applications, you would potentially have to change it in many places, probably missing one or more.

If you have an Oracle Client or Oracle Database Server installed, you may find the TNSNAMES.ORA file in the following location:

```
ORACLE_HOME/network/admin
```

Below is an example of a simple format of a *net service name* and connection descriptor and an example with actual values:

```

net_service_name=
  (DESCRIPTION=
    (ADDRESS=(protocol_address_information))
    (CONNECT_DATA=
      (SERVICE_NAME=service_name)))

# with values:
ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = 192.168.24.56) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
)

```

1.3 The WHAT-WHAT (TNSNAMES.ORA) in Action

As I have mentioned, you can use the *net service names* defined in the `TNSNAMES.ORA` file to tell the driver to which database you want to connect to. The database URL for this approach looks like this:

```
jdbc:oracle:thin:@NET_SERVICE_NAME
```

where the *net service name* is an entry from our protagonist file. Before you can connect to Oracle using this approach, there is one more thing you have to do – you have to tell `JDBC` where to look for the `TNSNAMES.ORA` file.

To do that, you set the `oracle.net.tns_admin` property to the location of the folder containing your `TNSNAMES.ORA` file, accessible from your machine:

```

System.setProperty(
  "oracle.net.tns_admin",
  "C:/app/product/11.2.0/client_1/NETWORK/ADMIN");

```

And... that's it! All you have to do now is create a connection with the chosen *net service name* and *let the records out (or in)*!

If you don't have `TNSNAMES.ORA` file, you can create it yourself and save it in location of your choosing and point to it in above `setProperty` method.

Below is an example of utilizing `TNSNAMES.ORA` file entry in Java to connect to Oracle database:

```

public class UsingTNSNAMESExample {

    public static void main(String[] args) throws Exception {
        // tell the driver where to look for the TNSNAMES.ORA file
        System.setProperty( "oracle.net.tns_admin",
            "C:/app/product/11.2.0/client_1/NETWORK/ADMIN");

        // ORCL is net service name from the TNSNAMES.ORA file
        String dbURL = "jdbc:oracle:thin:@ORCL";

        // load the driver
        Class.forName("oracle.jdbc.OracleDriver");

        Connection conn = null;
        Statement stmt = null;
    }
}

```

```

try {
    conn = DriverManager.getConnection(dbURL, "your_username", "your_password");

    stmt = conn.createStatement();

    ResultSet rs = stmt.executeQuery("SELECT dummy FROM dual");

    if (rs.next()) {
        System.out.println("Dummy is equal to: " + rs.getString(1));
    }
} catch (Exception e) {
    e.printStackTrace();
}
finally {
    if (stmt != null) try { stmt.close(); } catch (Exception e) {}
    if (conn != null) try { conn.close(); } catch (Exception e) {}
}
}
}

```

The following string was printed to the standard output:

```
Dummy is equal to: X
```

I also enclose once again the content of a simple `TNSNAMES.ORA` file:

```

ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = 192.168.24.56) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
)

```

2 Further Reading & Useful Links

Below you'll find links to some useful resources on the Internet about topics covered in this article:

- http://docs.oracle.com/cd/B19306_01/java.102/b14355/urls.htm#BEIDIJCE
About datasources and URLs – Oracle Documentation
- http://docs.oracle.com/cd/E11882_01/network.112/e10835/tnsnames.htm
About TNSNAMES – Oracle Documentation
- <http://www.orafaq.com/wiki/JDBC> – About JDBC - OracleFAQ

I hope you enjoyed my article. If you have found any errors in it (even typos), you think that I haven't explained anything clearly enough or you have an idea how I could make the article better – please, do not hesitate to contact me, or leave a comment.