

JDBC

4 types of JDBC drivers

Type 1: JDBC-ODBC bridge

It is used for local connection.

ex) 32bit ODBC in windows

Type 2: Native API connection driver

It is connected by the Native Module of dependent form of h/w like .dll or .so.

ex) OCI driver for local connection to Oracle

Type 3: Network connection driver

Type 4: Database Protocol driver

It is independent from h/w because this driver is in Java.

ex) thin driver for local/global connection to Oracle

*** Type1, type 2, and type 3 are usually used.

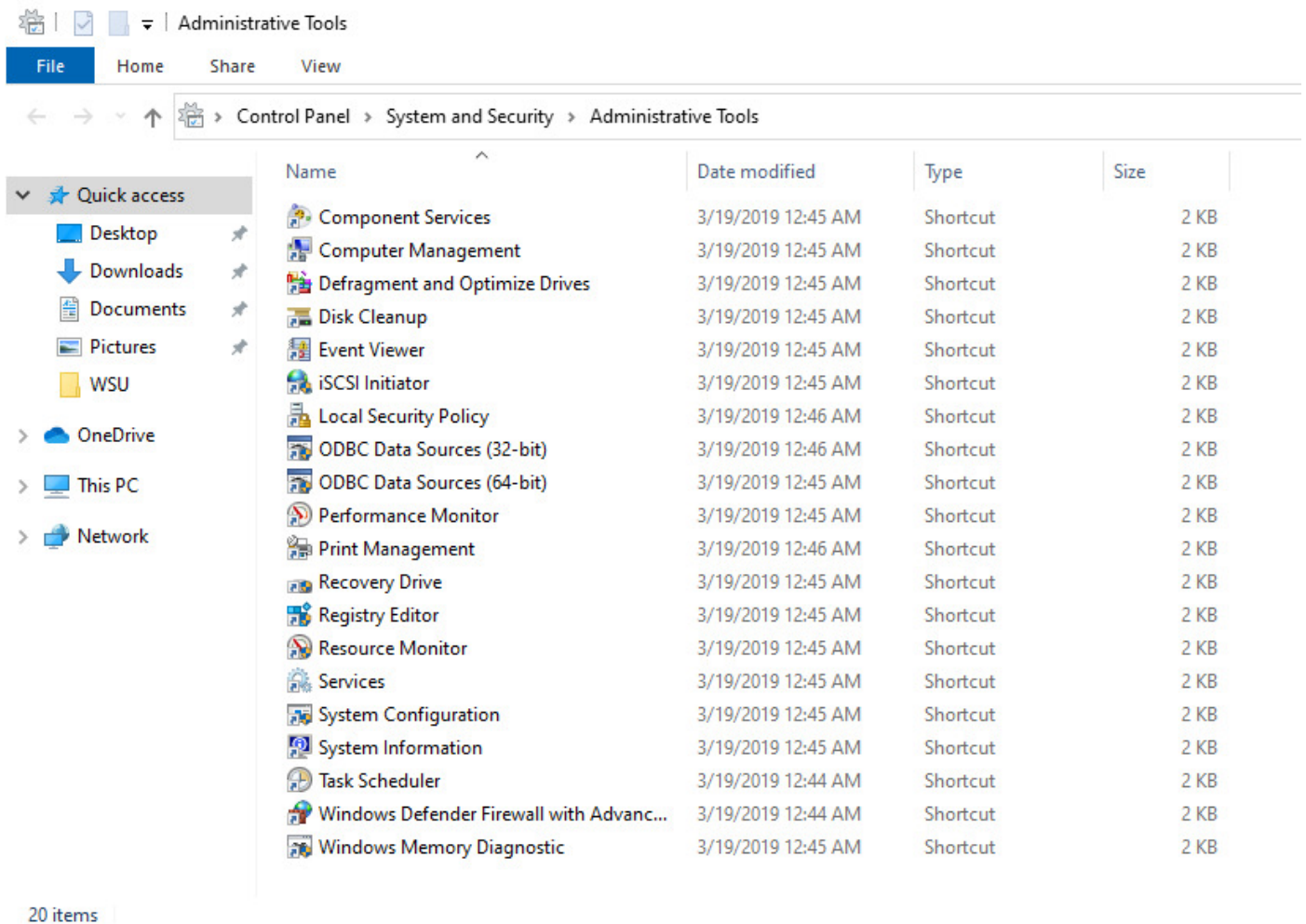
Examples of type1 and type 4 are available here.

Example of JDBC-ODBC Bridge

OS: Windows 10

DBMS: MS Access 2019

1. Open control panel and go to Administrative tools.



2. Right click on Data Sources (ODBC).

In properties, change the following and click 'OK'

Target field to:

%SystemRoot%\SysWOW64\odbcad32.exe

From:

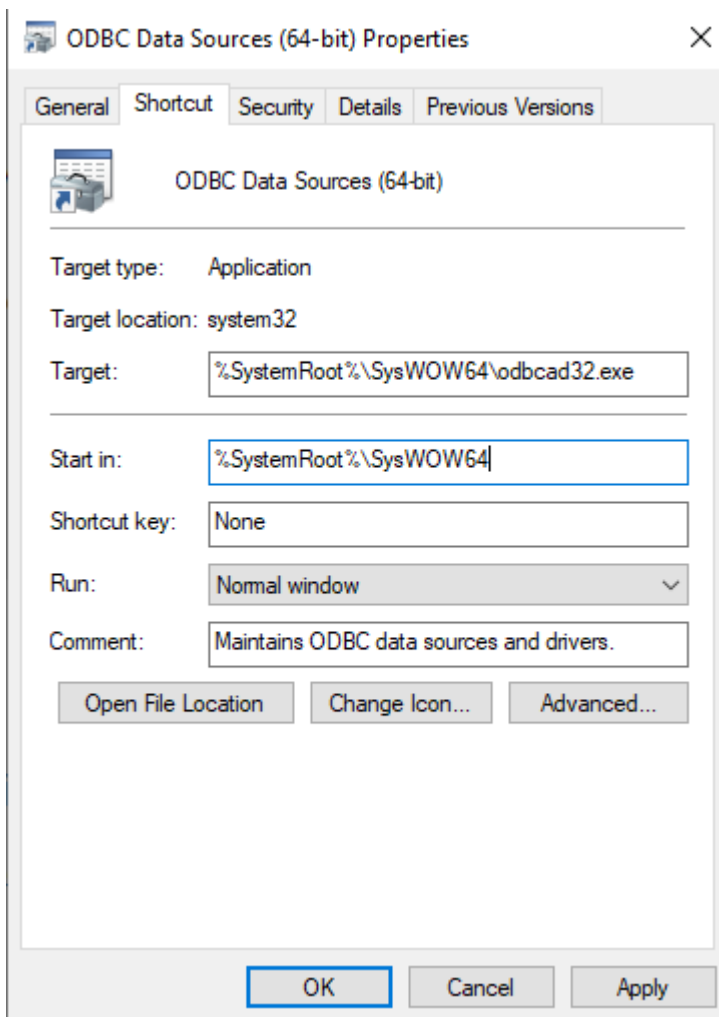
%SystemRoot%\System32\odbcad32.exe

Start In to:

%SystemRoot%\SysWOW64

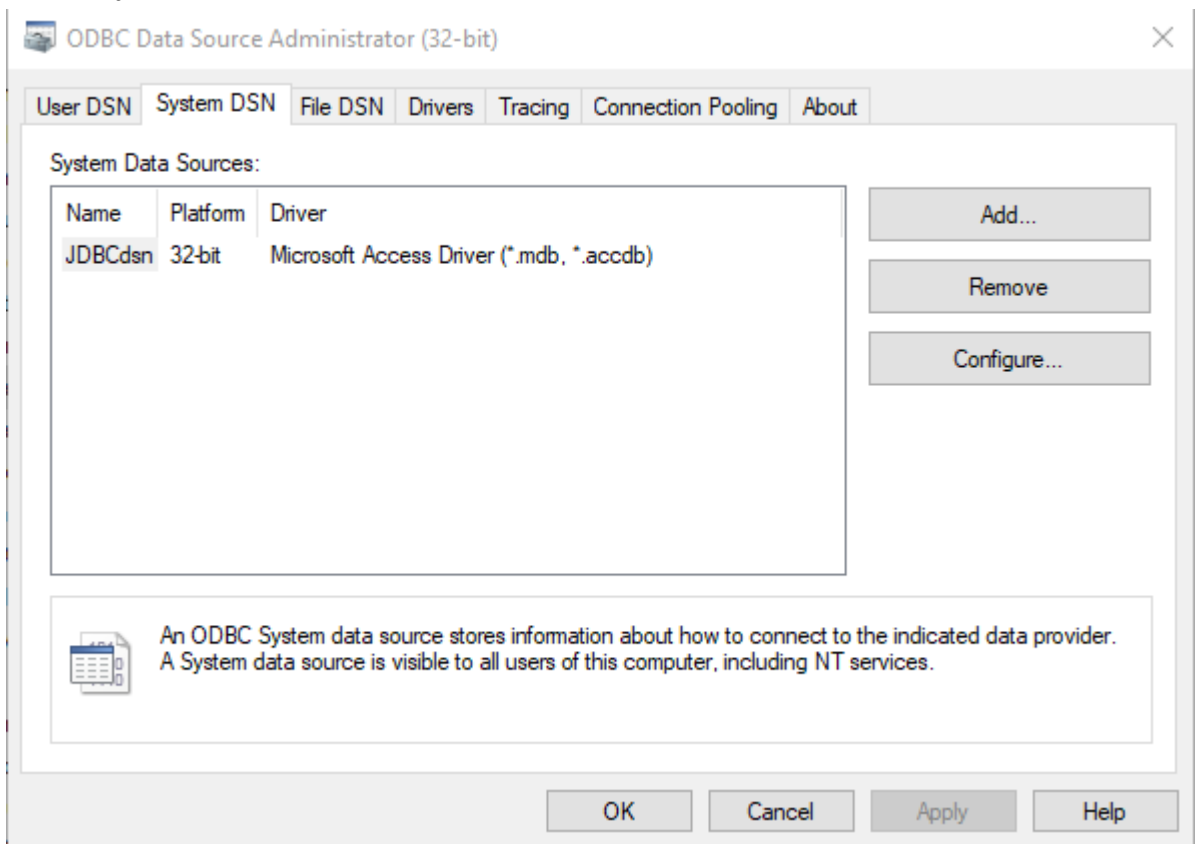
From:

%SystemRoot%\System32



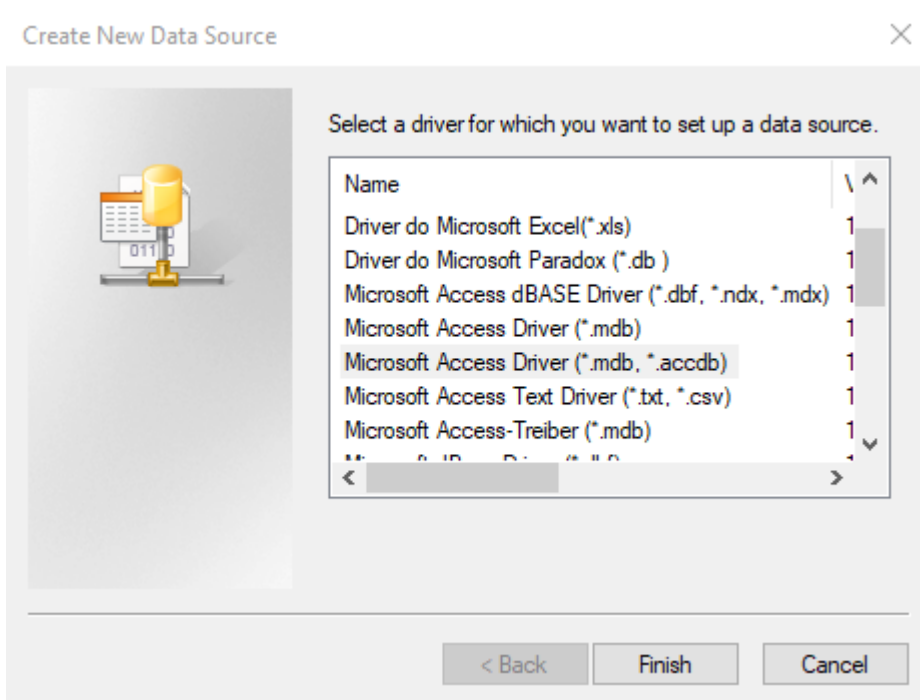
3. Double click on ODBC Data Sources (32-bit) or ODBC Data Sources (64-bit), depending on your installation of Microsoft Office.

Go to System DSN.



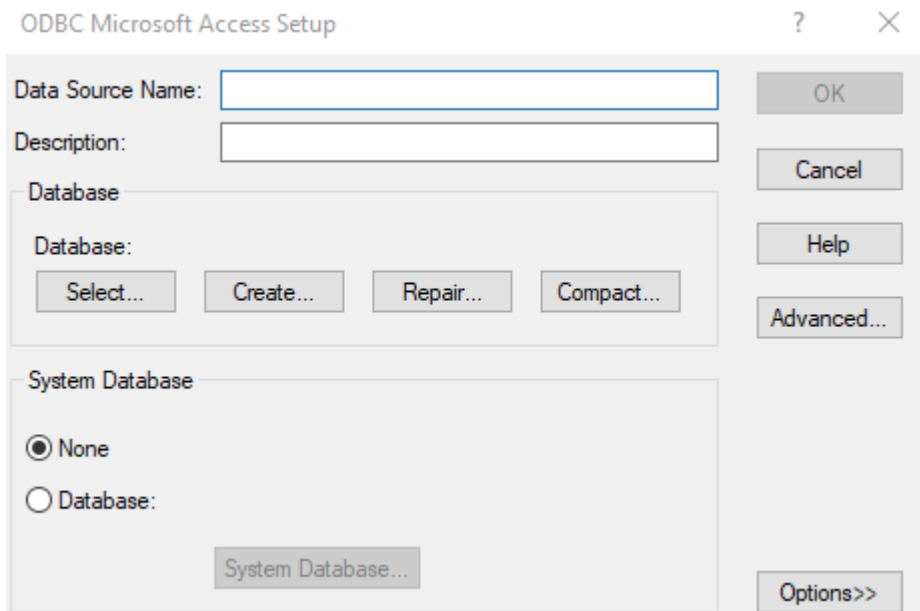
Click on Add button.

4. Then...



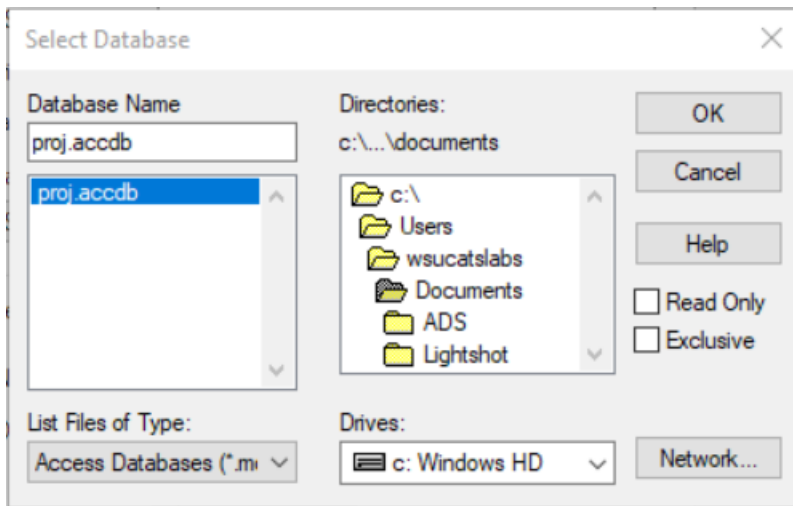
Click Microsoft Access Driver (*.mdb, *.accdb) and click on Finish button.

5. Then...



Insert your own Data Source Name (this is the name you will be using in the Java code to connect to the database, so ideally try to keep the database name and the DSN name to be the same) and click on Select button.

6. Then...



Choose your Database Access file like above and click OK button.

*** Note: before these procedures, we must have a Database Access file.

We can make this file by using MS Access.

7. Now, we can test our JDBC program with MS Access.

Here is a simple Java code that executes a SELECT statement.

```
import java.sql.*;

public class connect_msaccess
{
    public static void main(String[] args)
    {

        int i;
        Connection conn = null;

        // register jdbc driver
        try{
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            System.out.println("driver loaded...");
```

```

    } catch(ClassNotFoundException e) {
        System.out.println(e);
    }

    // connect to DB
    try{
        conn = DriverManager.getConnection("jdbc:odbc:JDBCdsn");
    } catch(SQLException se) {
        System.out.println(se);
    }
    System.out.println("connection is successful!!!");

    try{
        String selectSQL = "select ID, NAMES, ADDRESS from tb_address";
        Statement stmt = conn.createStatement();
        ResultSet rset = stmt.executeQuery(selectSQL);

        while(rset.next()){
            System.out.println("ID: " + rset.getString(1) + " NAMES: " +
                rset.getString(2) + " ADDRESS:" +
                rset.getString(3));
        }

        stmt.close();
    } catch(SQLException se) {
        System.out.println(se);
    }
}
}
}

```

8. Here is the result.

```

connect_msaccess x
"C:\Program Files (x86)\Java\jdk1.7.0_80\bin\java.exe" ...
driver loaded...
connection is successful!!!
ID: 1 NAMES: Tom ADDRESS:LA
ID: 2 NAMES: John ADDRESS:Dayton, OH
ID: 3 NAMES: David ADDRESS:Columbus, OH

Process finished with exit code 0

```

Example of thin driver

- OS: Window 10
- DBMS: Oracle 11g with SQL Plus (available in 152 RC labs A/C and 346 RC)

1. Download a suitable thin driver from Oracle website

In the Russ labs, it exists in C:\DevSuiteHome\jdbc\lib

The drivers are usually specified as: classes111.zip for JDK 1.1.x and classes12.zip for JDK 1.2.x or more

2. Set your path (environment variable)

In the Russ labs, you need to add to classpath (Sometimes you might have to restart the system):

```
classpath= ;C:\DevSuiteHome\jdbc\lib\classes12.zip
```

Java path precedes Oracle path.

3. Test with a simple Java code.

Before implementing the example Java code shown below, you should create a table, tb_address, by using SQL Plus in Oracle and insert some values.

For Oracle thin driver, you can use the following in the Java code:

```
jdbc:oracle:thin:@//<HOST>:<PORT>:<SID> - url would look like this for SQL Plus, to make connection
```

```
jdbc:oracle:thin:@<HOST>:<PORT>:<SID> - url would look like this in general, for making connection
```

```
oracle.jdbc.driver.OracleDriver - for loading driver
```

To find the host and port, follow the path (in Russ labs):

```
C:\DevSuiteHome\NETWORK\ADMIN
```

And open the tnsnames.ora file in Notepad, then you will find hostname and port.

For the Russ labs,
HOST is cseora.cs.wright.edu
PORT is 1521
SID (Service Name) is cseora11

For example, this is how to make a connection in Java with SQL Plus (in Russ labs):

```
conn = DriverManager.getConnection("jdbc:oracle:thin:@//cseora.cs.wright.edu/cseora11", "scott", "tiger");
```

```
// scott: username of Oracle database
```

```
// tiger: password of Oracle database
```

*** Here is a simple Java code

```
import java.sql.*;

public class connect_thin
{
    public static void main(String[] args)
    {

        int i;
        Connection conn = null;

        // register jdbc driver
        try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
            System.out.println("driver loaded...");
            // in order to use Oracle thin/oci driver

        } catch(ClassNotFoundException e) {
            System.out.println(e);
        }

        // connect to DB
        try{

            conn = DriverManager.getConnection("jdbc:oracle:thin:@//cseora.cs.wright.edu/cseora11", "scott", "tiger");

        } catch(SQLException se) {
            System.out.println(se);
        }
        System.out.println("connection is successful!!!");

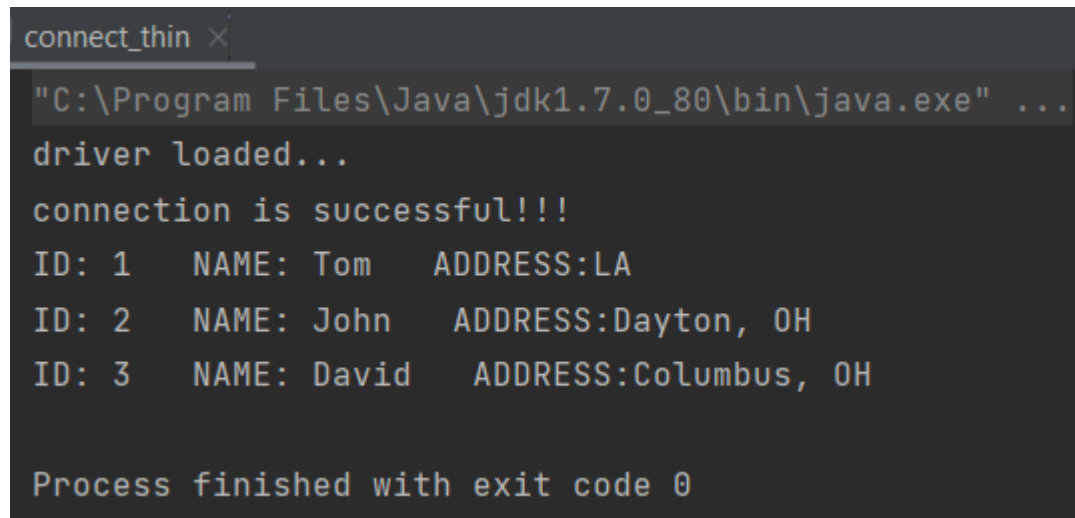
        try{
            String selectSQL = "select ID, NAME, ADDRESS from tb_address";
            Statement stmt = conn.createStatement();
```

```
ResultSet rset = stmt.executeQuery(selectSQL);

while(rset.next()){
    System.out.println("ID: " + rset.getString(1) + " NAME: " +
        rset.getString(2) + " ADDRESS:" +
        rset.getString(3));
}

stmt.close();
} catch(SQLException se) {
    System.out.println(se);
}
}
}
```

4. Result of a simple Java code.



```
connect_thin x
"C:\Program Files\Java\jdk1.7.0_80\bin\java.exe" ...
driver loaded...
connection is successful!!!
ID: 1 NAME: Tom ADDRESS:LA
ID: 2 NAME: John ADDRESS:Dayton, OH
ID: 3 NAME: David ADDRESS:Columbus, OH

Process finished with exit code 0
```