# How-to: Direct DB Access from R using ODBC

How to install the Oracle Client and set up ODBC connection for use in programs like R or Python in Windows 10 using DSN. You will need admin privileges to set this up. In many cases that will involve having IT perform the install. The DSN setup will allow the data sources to more easily be used in other programs. This document will focus on using the connection in R after setting up the source.

Contact Robert Ryznar (rryznar@psmfc.org) or staff to obtain account information. When copying and pasting be careful of the quotation marks. Word processors often stylize the quotation marks so they look curved (e.g. “the” vs "the"). R and Oracle are usually tripped up by the curved ones.

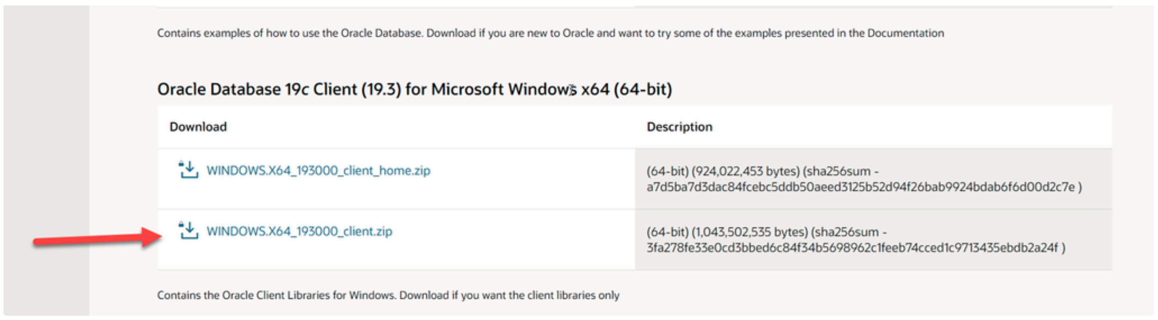
Oracle SQL is generally not case sensitive, but when a phrase has quotes around it will look for the exact way it appears in the quotes. For example, the R command to view the list of tables through RODBC is dbListTables(channel, schema="COUNCIL"). Ignore channel for now, but the schema needs to be specified as upper case. Using schema="council" or “Council” will return a result of "character(0)." This means there were no results, and often indicates something is spelled incorrectly or has a case issue.

As of May 2021, the current Long Term Release of the Oracle Database Client is version 19.3. Almost all users will want the 64 bit version of the client, which is available at https://www.oracle.com/database/technologies/oracle19c-windows-downloads.html

About 2/3rds of the way down the page is a section called “Oracle Database 19c Client (19.3) for Microsoft Windows x64 (64-bit).” Choose the version without home in the name.

A copy of the file is also available in our ShareFile system to make it easier to find and prevent the need to register at the Oracle site. The link is https://psmfc.sharefile.com/d-s2f4f047578684c7e8a5a3eb20cff3fc4

Contact nleuthold@psmfc.org if you are unable to access this file.



Contains examples of how to use the Oracle Database. Download if you are new to Oracle and want to try some of the examples presented in the Documentation

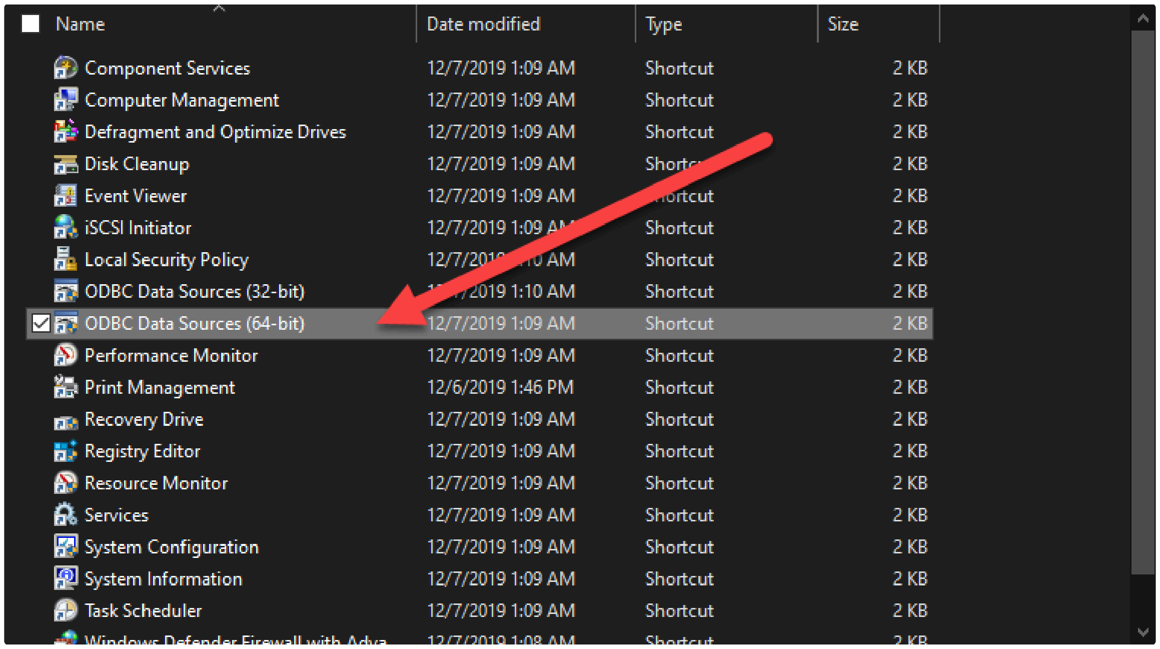
**Oracle Database 19c Client (19.3) for Microsoft Windows x64 (64-bit)**

| Download | Description |
|---|---|
| WINDOWS.X64_193000_client_home.zip | (64-bit) (924,022,453 bytes) (sha256sum - a7d5ba7d3dac84fcebc5ddb50aeed3125b52d94f26bab9924bdab6f6d00d2c7e ) |
| WINDOWS.X64_193000_client.zip | (64-bit) (1,043,502,535 bytes) (sha256sum - 3fa278fe33e0cd3bbed6c84f34b5698962c1feeb74cced1c9713435ebdb2a24f ) |

Contains the Oracle Client Libraries for Windows. Download if you want the client libraries only

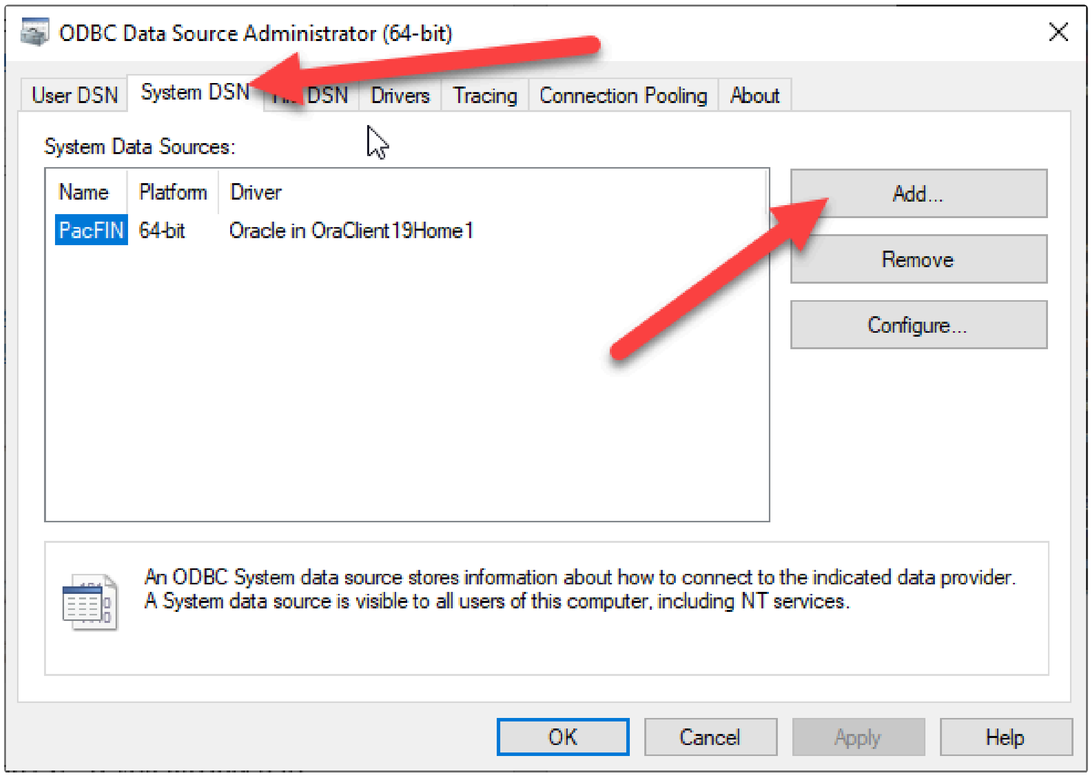
1. Once the download is complete unzip the file and run the setup.exe. If you unzipped to your c drive the path would be C:\WINDOWS.X64_193000_client\client\setup.exe
   a. Select Administrator option
   b. Follow the prompts for a default install
2. Once the install is complete you will need to replace the tnsname.ora file. This file specifies the connection information for the databases. Please contact Robert Ryznar (RRyznar@psmfc.org) or staff to get the correct version of the file.
   a. The default install will place the tnsnames.ora file in C:\app\client\nleuthold\product\19.0.0\client_1\network\admin
      i. This will be your user name
      ii. This might read 19.3.0
      iii. This is likely client
   b. Rename the existing copy of tnsnames.ora to tnsnames.old and copy the tnsnames.ora file obtained from staff to the directory
3. Firewall rules – May need to contact

4. Users not interested in setting up access to other programs should be able to connect through Oracle SQL Developer
5. To make the database connection available to other programs we will need to setup the connection as an ODBC Data Source.
   a. Click the start button and type “Windows Administrative Tools” and click name. It should show up after you are a couple letters into the name.
   b. This will open a popup directory scan down and double click “ODBC Data Sources (64-bit)”



| Name | Date modified | Type | Size |
|---|---|---|---|
| Component Services | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| Computer Management | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| Defragment and Optimize Drives | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| Disk Cleanup | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| Event Viewer | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| iSCSI Initiator | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| Local Security Policy | 12/7/2019 1:10 AM | Shortcut | 2 KB |
| ODBC Data Sources (32-bit) | 12/7/2019 1:10 AM | Shortcut | 2 KB |
| ODBC Data Sources (64-bit) | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| Performance Monitor | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| Print Management | 12/6/2019 1:46 PM | Shortcut | 2 KB |
| Recovery Drive | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| Registry Editor | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| Resource Monitor | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| Services | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| System Configuration | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| System Information | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| Task Scheduler | 12/7/2019 1:09 AM | Shortcut | 2 KB |
| Windows Defender Firewall with Adva | 12/7/2019 1:08 AM | Shortcut | 2 KB |

   c. A popup will open. Click the “System DNS” Tab and click Add…



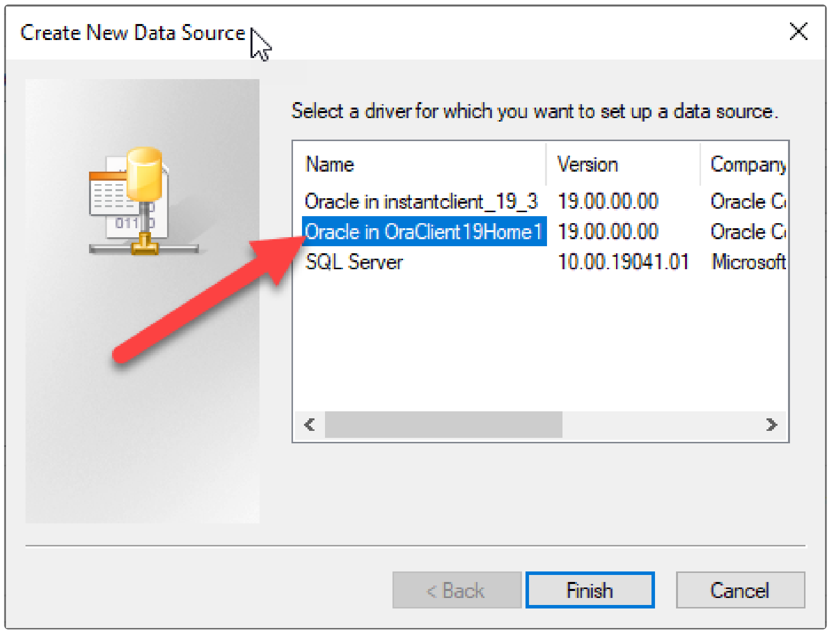
ODBC Data Source Administrator (64-bit)

System Data Sources:

| Name | Platform | Driver |
|---|---|---|
| PacFIN | 64-bit | Oracle in OraClient19Home1 |

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users of this computer, including NT services.

   d. The next popup will vary some based on the services you have installed already. Look for the version titled “Oracle in OraClient19Home.” Do not use a version with instant client in the title. Click finish and a configuration window will pop up



Create New Data Source

Select a driver for which you want to set up a data source.

| Name | Version | Company |
|---|---|---|
| Oracle in instantclient_19_3 | 19.00.00.00 | Oracle C |
| Oracle in OraClient19Home1 | 19.00.00.00 | Oracle C |
| SQL Server | 10.00.19041.01 | Microsoft |

e. In the config window enter

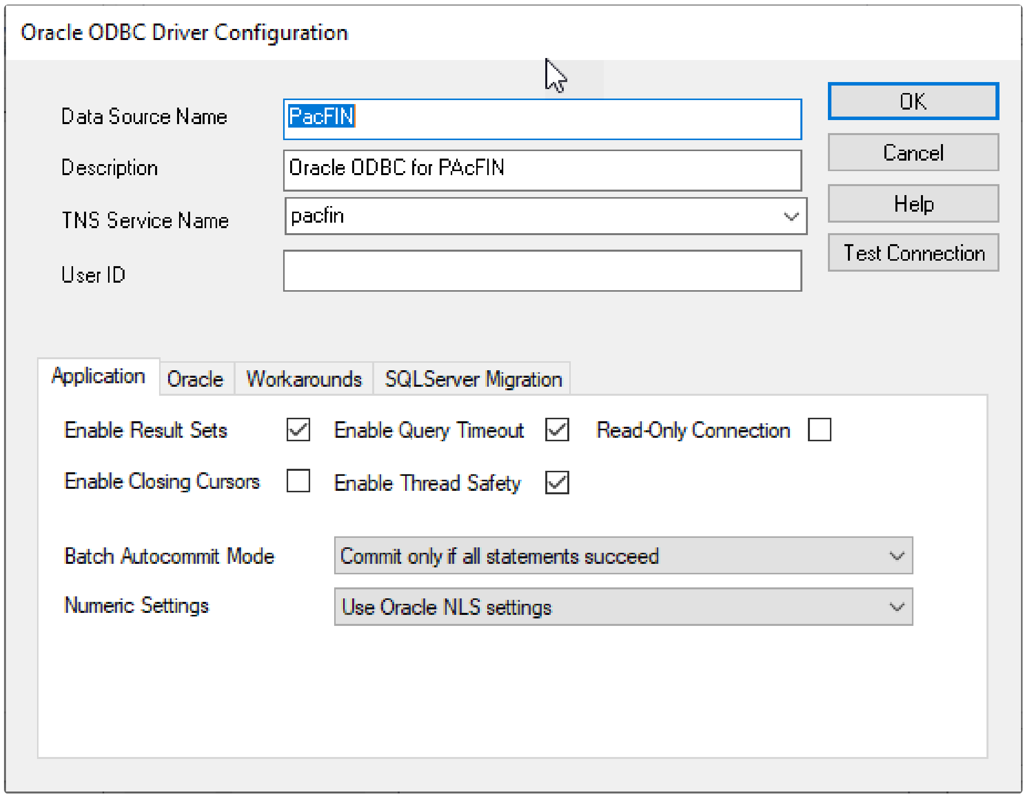
i. Data Source Name – AKFIN, PacFIN or RecFIN based on service you are accessing. You can add a longer name if that is helpful to you

ii. Description – up to you, but something to help you identify what it is

iii. TNS server name is either akfin for Alaska data, or PacFIN for west coast commercial or recreational data (PacFIN and/or RecFIN)

iv. I leave the user ID blank. You can use scripts in R to automatically load your user ID and password

v. Click test and enter your user ID and password. If it passes click OK. If not please note error and contact staff to figure out the error



Oracle ODBC Driver Configuration

Data Source Name: PacFIN

Description: Oracle ODBC for PAcFIN

TNS Service Name: pacfin

User ID:

Enable Result Sets ✓ Enable Query Timeout ✓ Read-Only Connection ☐

Enable Closing Cursors ☐ Enable Thread Safety ✓

Batch Autocommit Mode: Commit only if all statements succeed

Numeric Settings: Use Oracle NLS settings

1. Now to use the connection in R

a. There are several libraries that could be used. This document will show examples for 1) RODBC and 2) odbc. Either library can be used, but typically AKFIN users have used odbc while PacFIN users have typically used RODBC, so the examples are setup in that manner.

b. For whichever one you choose to use, install the package/library through R so that any needed supporting packages are also installed.

c. Library("RODBC") or library ("odbc")

d. Henceforth commands for RODBC will have a RODBC: before them and the commands for odbs will have an odbc: before them

e. get data about the session
    i. sessionInfo() works for either - not part of ODBC

f. Check available ODBC connections
    i. RODBC: odbcDataSources(type=c('all'))
    ii. odbc: odbcListDataSources()

g. The entry you previously created (step e) should be one of the listed options.

h. Setup connection and get basic info
    i. RODBC:
        1. channel<- odbcConnect(dsn="PacFIN",uid= "username",pw= "your pass",believeNRows=FALSE)
        2. the dsn will be the name you gave the service in e (i)
        3. substitute your user name and password
        4. odbcGetInfo(channel)
        5. shows connection driver info
    ii. odbc:
        1. channel<- dbConnect(odbc::odbc(),dsn="AKFIN",uid="username",pwd="your pass"))
        2.

2. To see tables you can access
    a. RODBC:
        i. sqlTables(channel,schema="PACFIN_MARTS")
    b. odbc:
        i. dbListTables(channel, schema = "COUNCIL")
3. example query of PacFIN table
    a. RODBC:
        i. set channel - channel<- odbcConnect(dsn="PacFIN",uid="user name",pw="password",believeNRows=FALSE)
        ii. query to select ifq sablefish - sabl <- sqlQuery(channel,"select PACFIN_PORT_CODE,  LANDING_YEAR, PACFIN_GROUP_GEAR_CODE, sum(LANDED_WEIGHT_LBS)

```
   from PACFIN_MARTS.COMPREHENSIVE_FT
   where IS_IFQ_LANDING='T' and PACFIN_PORT_CODE in  ('MRO','AVL','MNT','MOS','PRN','SF')
   group by PACFIN_PORT_CODE, LANDING_YEAR,              PACFIN_GROUP_GEAR_CODE")
```

sabl – to see output

1. ODBC
    a. set channel - channel<- dbConnect(dsn="AKFIN",uid="user name",pw="password")
    b. sabl <- dbGetQuery(channel, "SELECT BCA.AKFIN_YEAR as year, BCA.FMP_AREA, BCA.AGENCY_SPECIES_CODE, BCA.MANAGEMENT_PROGRAM_CODE, sum(BCA.WEIGHT_POSTED ) WEIGHT_POSTED

```
                                FROM council.COMPREHENSIVE_BLEND_CA BCA
   WHERE BCA.AKFIN_YEAR between 2018 and 2020
      and BCA.AGENCY_SPECIES_CODE = 710
      and BCA.MANAGEMENT_PROGRAM_CODE = 'IFQ'
      and BCA.FMP_AREA != 'INSD'
      GROUP by BCA.AKFIN_YEAR, BCA.FMP_AREA,
      BCA.AGENCY_SPECIES_CODE,
          BCA.MANAGEMENT_PROGRAM_CODE
      ORDER BY 1, 2, 3" )
```

sabl to view output

1. Close the data connection when done
   a. RODBC:
      i. close(channel)
   b. odbc:
      i. dbDisconnect(channel)

---

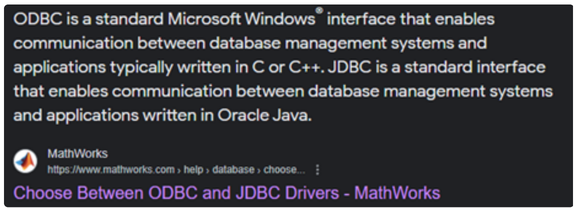
## JDBC Alternative

If the method above produces poor performance (e.g., extended import times for large data sets) the Java Database Connectivity (JDBC) may provide an alternative. The following guide has been adapted from detailed instructioned created by Ben Lincoln (RDI) here:

- "N:\Shared Folders\recfin\projects\mrfss_legacy_migration\JDBC in RStudio for basic scripting with database objects in Oracle.docx"
- "N:\Shared Folders\recfin\projects\mrfss_legacy_migration\test_jdbc_plus_diff.R"

### Java Virtual Machine (JVM)

R is a language based on the Java Virtual Machine (JVM), and as such, JDBC may provide improved performance compared to ODBC.



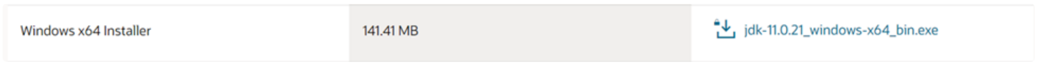
ODBC is a standard Microsoft Windows® interface that enables communication between database management systems and applications typically written in C or C++. JDBC is a standard interface that enables communication between database management systems and applications written in Oracle Java.

MathWorks
https://www.mathworks.com › help › database › choose...

Choose Between ODBC and JDBC Drivers - MathWorks

### Java Development Kit (JDK)

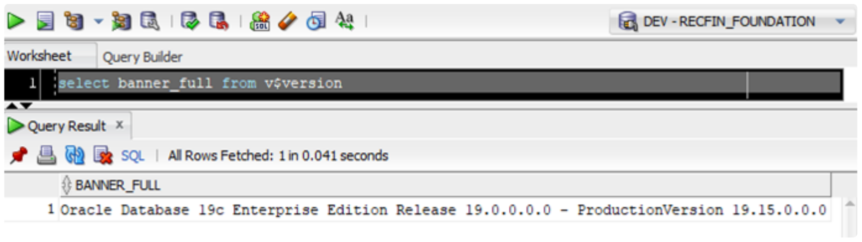
JDBC requires the Java Development Kit (JDK). As of this writing, the current version is JDK 11.  You can download JDK here.



| | | |
|---|---|---|
| Windows x64 Installer | 141.41 MB | jdk-11.0.21_windows-x64_bin.exe |

### Oracle JDBC Driver

Install the Oracle driver specific to the Oracle version currently being used by PSMFC. As of this writing, this is Oracle 19c.

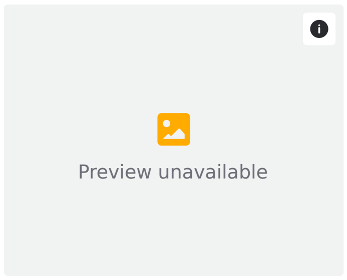


```
select banner_full from v$version
```

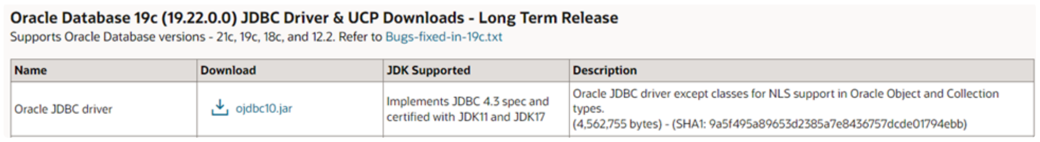
All Rows Fetched: 1 in 0.041 seconds

```
  BANNER_FULL
1 Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - ProductionVersion 19.15.0.0.0
```

Note that future Oracle drivers are backward compatible, but RecFIN testing found the version for Oracle 19c was marginally faster. In the event PSMFC upgrades its Oracle system to another version or if another Oracle system and/or version is used, it is advised  to update the version of the jar file used for JDBC to correspond to the respective Oracle version being used.



The JDBC driver jar files can be found here.



**Oracle Database 19c (19.22.0.0) JDBC Driver & UCP Downloads - Long Term Release**

Supports Oracle Database versions - 21c, 19c, 18c, and 12.2. Refer to Bugs-fixed-in-19c.txt

| Name | Download | JDK Supported | Description |
|---|---|---|---|
| Oracle JDBC driver | ojdbc10.jar | Implements JDBC 4.3 spec and certified with JDK11 and JDK17 | Oracle JDBC driver except classes for NLS support in Oracle Object and Collection types. (4,562,755 bytes) - (SHA1: 9a5f495a89653d2385a7e8436757dcde01794ebb) |

## RStudio and R Installation

If not already installed, download and install R and RStudio.

The R software download can be found here.

The RStudio download can be found here.

## RStudio script configuration for JDBC

### JAVA_HOME

With JDK, there is a concept of JAVA_HOME, which can be configured at the OS level, but RStudio allows for configuration using script code from the RStudio environment. Here is an example of that:

```r
# Set JAVA_HOME, set max. memory, and load RJava library
# install jdk 11

Sys.setenv(JAVA_HOME='C:/Program Files/Java/jdk-11')
options(java.parameters="-Xmx2g")
```

Please note the forward slashes. This folder is where the jdk 11 files is installed. The “-Xmx2g” parameter denotes the amount of memory the process has access to.

### RJDBC

Run the script below to install the RJDBC package which includes three required libraries: rJava, RJDBC, and DBI.

```r
# run this line ONCE to install rJava, RJDBC, DBI
install.packages("RJDBC")
```

### JDBC

Run the script below to create the connection to the jdbc driver. The path referenced in the script is the directory location of the JDBC driver jar file (reference above). Update the script as needed based on the directory location on your machine where the jar file is stored.

```r
# Create connection driver and open connection
# you control the location of the jar file.  I have it for you.
jdbcDriver <- JDBC(driverClass="oracle.jdbc.OracleDriver",
classPath="c:/Users/cbaer/sandbox/Oracle_JDBC/ojdbc10.jar")
```

### dbConnect

Run the script below to connect. In general, the pattern logic is as follows. Make sure to update the `username` and `password`.

```
jdbcConnection <- dbConnect(jdbcDriver, "jdbc:oracle:thin:@//pacfindb.psmfc.org:2045/pacfin.psmfc.org", "username", password)
```

### dbGetQuery

The call to this function is very similar to the ODBC analog, sqlQuery.  It returns a data frame, which can then be manipulated in the R environment. As an example:

```
DCRFS_10963 <- dbGetQuery(jdbcConnection, "select * from RECFIN_LEGACY.DCRFS_10963")
DCRFS_10963_V <- dbGetQuery(jdbcConnection, "select * from RECFIN_LEGACY.DCRFS_10963_V")
```