# Accessing an Oracle Database from SAS on Microsoft Windows

On Microsoft Windows, you have an option to access an Oracle database from SAS. You can use SAS/ACCESS Interface to Oracle, SAS/ACCESS Interface to ODBC, or SAS/ACCESS Interface to OLE DB.

Submitting the following code from within SAS displays in the SAS log window all products currently licensed at your site:

```
proc setinit noalias;
run;
```

Based on this information, you can choose which if the following methods you will use to access your Oracle data.

### *SAS/ACCESS Interface to Oracle:*

Software Requirements:

- Verify that the SAS/ACCESS Interface to Oracle is installed on your machine.
  - Browse to !SASROOT\Access\Sasexe and verify existence of sasioora.dll.
- Verify that the SAS/ACCESS Interface to Oracle is licensed.
  - Submit PROC SETINIT code from above example. "SAS/ACC – ORACLE" should appear in the SAS log.
- Verify that the Oracle SQL*Net Client is installed on the machine.
  - The Oracle DBA at your site can help on installing the Oracle SQL*Net Client to your machine.

- For additional information on system requirements for the SAS/ACCESS Interface to Oracle product please refer to the system requirements at http://support.sas.com/installcenter.

Once you have met the above software requirements, you can connect to an Oracle database from SAS by using one of the following methods:

- LIBNAME statement using the ORACLE engine (provides direct and dynamic access to the Oracle data).

```
libname oralib oracle user=scott password=xxxxx
path=V2o7223 preserve_tab-names=yes;
```

Note: The values for user=, password=, and path= are specific to Oracle. The path= statement is the Oracle alias name that is defined in the Oracle TNSNAMES.ORA file.

- The SQL Procedure Pass-Through Facility enables you to pass Oracle SQL statements to an Oracle database for processing.

```
proc sql;
connect to oracle(user=scott password=tiger
path=dark_o8150);
create table test1 as
select * from connection to Oracle
(select * from Dept where rownum <=5);
quit;
```

- Use of the ACCESS and DBLOAD procedures should be restricted to compatibility with V6 applications and naming conventions. These are not discussed in this TS note but are documented in the SAS Online Doc, Version 8.

### *SAS/ACCESS Interface to ODBC*

The SAS/ACCESS Interface to ODBC software provides an interface between SAS and any ODBC-compliant application.

Software Requirements:

- Verify that the SAS/ACCESS Interface to ODBC is installed on your machine.
  - Browse to !SASROOT\Access\Sasexe and verify existence of sasioodb.dll.
- Verify that the SAS/ACCESS Interface to ODBC is licensed.
  - Submit proc setinit code from above example. SAS/ACC – ODBC should appear in the SAS Log.
- Verify that the ORACLE ODBC driver for the installed on your machine.
  - Open the Data Sources (ODBC) from the Microsoft Windows Control Panel.
  - On the Drivers tab verify that the correct driver is installed.
  - Define an ODBC Data Source to your ORACLE database.

Once you have met the above software requirements, you can connect to an Oracle database from SAS by using one of the following methods:

- LIBNAME statement with the ODBC engine (provides direct and dynamic, access to the Oracle data).

```
libname odbclib odbc dsn=ora1 user=scott
password=password;
```

The SQL Procedure Pass-Through Facility enables you to pass Oracle SQL statements to an Oracle database via the Oracle ODBC driver.

```
proc sql;
connect to odbc(dsn=oracle1 uid=scott password=tiger);
create table test1 as
select * from connection to odbc
(select * from Dept);
quit;
```

### *SAS/ACCESS Interface to OLE DB*

SAS/ACCESS Interface to OLE DB software accesses data from Oracle through the Oracle OLEDB data providers. The Microsoft OLE DB API provides access to data, which can be in many forms.

Software Requirements:

- Verify that the SAS/ACCESS Interface to OLEDB is installed on your machine.
  - Browse to !SASROOT\Access\Sasexe and verify existence of sasioole.dll.
- Verify that the SAS/ACCESS Interface to OLEDB is licensed.
  - Submit proc setinit code from above example. SAS/ACC – OLEDB should appear in the SAS Log.
- Verify that the ORACLE Data Provider has been installed.
  - Should be installed if the Oracle SQL*Net Client is installed on your machine.

Once you have met the above software requirements, you can connect to an Oracle database from SAS by using one of the following methods:

- LIBNAME statement with the OLEDB engine (provides direct and dynamic access to the Oracle data.via the Oracle OLE DB provider).

```
libname oradb oledb init_string="Provider=MSDAORA.1;
Password=tiger;User ID=scott;
Data Source=dark_o8150;
Persist Security Info=True";

proc print data=mydblib.customers;
where state='CA';
run;
```

- The SQL Procedure Pass-Through Facility enables you to pass Oracle SQL statements to an Oracle database for processing.

```
proc sql;
connect to oledb(init_string="Provider=MSDAORA.1;
Password=tiger;User ID=scott;
Data Source=dark_o8150;Persist Security
Info=True");
%put &sysdbmsg;
```

```
create table work.test as
select * from connection to oledb
(Select * from EMP);
quit;
```

Note: If you are unsure of the syntax for the connection options, use the method listed below to perform a prompted connection and obtain the connection information by using the MACRO SYSDBMSG:

```
 libname oradb oledb;
%put &sysdbmsg;
```

You will be prompted to select the datasource name from the Data Link Properties window. The macro variable &SQLDBXMSG will display the INIT_STRING information in the SAS Log window.

The SAS log will contain the following information:

```
OLEDB: Provider=MSDAORA.1;Password=tiger;User
ID=scott;Data Source=dark_o8150;Persist
Security Info=True;
```

Now, highlight and copy the entire string , excluding the OLEDB: and paste it after the init_string option, as shown below:

```
libname oradb oledb
init_string="Provider=MSDAORA.1;
Password=tiger;User ID=scott;
Data Source=dark_o8150;Persist Security
Info=True";
```

Make sure you enclose the entire initialization string in quotes.