

TS-518D

***SAS/ACCESS® Guidelines for Connecting to
ORACLE® Databases in the UNIX® Environment***

***Technical Support Division
UNIX Systems Interface***

***SAS Institute, Inc.
SAS Campus Drive
Cary, N.C. 27513***

Abstract

Due to the increased interest of the SAS/ACCESS Interface to ORACLE product in the UNIX environment, this topic has been a popular discussion for a number of SAS users. This paper documents the necessary steps and clarifies the issues involved with connecting to ORACLE databases using SAS/ACCESS on UNIX platforms.

Introduction

The SAS/ACCESS Interface to ORACLE product provides support for the SQL Procedure Pass-Through Facility, the ACCESS Procedure and the DBLOAD Procedure.

The SAS/ACCESS product, in general, consists of three procedures: PROC ACCESS, PROC DBLOAD and PROC SQL Pass-Through. However, all three of these procedures may not be available for every relational database system (DBMS) on every UNIX platform. Users should be familiar with what procedures their SAS/ACCESS product supports for their platform. An addendum has been included for the SAS/ACCESS Interface to ORACLE support.

The SQLPLUS utility is available with ORACLE to provide users the capability to retrieve and create data stored in the ORACLE database. Users or the Database Administrator (DBA) should be familiar with this utility because it can make the SAS/ACCESS connection process much easier.

The discussion, necessary steps, and examples that follow were run on an HP-UX system using SAS 6.11 TS020, and demonstrate the use of the SQLPLUS utility and successful connections to databases using SAS/ACCESS.

Using the ORACLE SQLPLUS Utility

In order to connect to an ORACLE database locally or remotely from SAS, you must be able to connect to the same database outside of SAS. You can establish a connection outside of SAS by using the ORACLE SQLPLUS utility.

*Once a successful database connection is made using the SQLPLUS utility, you should be able to connect to the same database from within SAS with the **same** environment and userid. Below are some simple steps to verify whether you are able to connect to an ORACLE database outside of SAS successfully.*

- 1) *Invoke the ORACLE SQLPLUS utility as follows, where `usr_name`, `password` and `connect_string` are valid values for connecting to the database. If you do not know what these values are or should be, or have problems connecting with the values you have specified, you will need to contact your DBA or ORACLE Technical Support for assistance. If you are not able to connect to ORACLE using the SQLPLUS utility with the **exact** syntax demonstrated below, you will not be able to use SAS/ACCESS to connect to the database either.*

```
unix_prompt> sqlplus usr_name/password@connect_string
```

- 2) *Once you successfully connect to the database specified, you will receive information similar to the following. If you are having problems connecting to the database, you will need to contact your DBA or ORACLE Technical Support for assistance.*

```
SQL*Plus: Release 3.1.3.5.1 - Production on Tue Aug 20
12:57:07 1996
```

```
Copyright (c) Oracle Corporation 1979, 1994. All rights
reserved.
```

```
Connected to:
```

```
Oracle7 Server Release 7.1.4.1.0 - Production Release
With the distributed option
PL/SQL Release 2.1.4.0.0 - Production
```

```
SQL>
```

- 3) *To verify that you have been granted the appropriate privileges, submit an SQL query from within SQLPLUS, similar to what you plan to submit from within SAS. If you experience problems with privileges or submitting SQL from SQLPLUS, you will need to contact your DBA or ORACLE Technical Support for assistance. Below is a simple SQL `select` statement and its results (where `tbl_name` is a valid table name in the database that you are connected):*

```
SQL> select * from tbl_name;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT
7934	MILLER	CLERK	7782	23-JAN-82	1300		10
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7788	SCOTT	ANALYST	7566	10-FEB-88	3000		20

- 4) *You have just demonstrated that you can connect to the ORACLE database outside of SAS. Before you attempt to bring up SAS and try to connect using SAS/ACCESS, be sure that you use the **exact** `usr_name`, `password`, `connect_string` and `tbl_name` that you used with the ORACLE SQLPLUS utility.*

Using SAS/ACCESS Interface to ORACLE

Before you attempt to connect to an ORACLE database using SAS/ACCESS, you must be able to connect to the same database outside of SAS using the ORACLE SQLPLUS utility as demonstrated above. Once you have verified that you can connect to the database(s) using the SQLPLUS utility, you should be able to connect to the same database(s) from within SAS, assuming the following requirements have been met.

- *You are invoking SAS using the **exact** environment and userid used to invoke the SQLPLUS utility and connected to the database(s) successfully.*
- *You have successfully performed the SAS/ACCESS Interface to ORACLE Installation Instructions from APPENDIX B of the "Installation Instructions for the SAS System under UNIX Environments".*
- *Make sure that the `!SASROOT/sasexe/dbi/sasorav7` module along with all other ORACLE related modules in this directory have read and execute permissions. For dual SAS installations on SUN systems, this module can be found in `!SASROOT/solaris2/sasexe/dbi` for Solaris 2 platforms.*
- *Make sure the environment variable `$SASORA` is set to V6, V7 or V8 depending on the ORACLE version you intend to access. The following syntax is used to set variables:*

Bourne Shell: `export SASORA=V7`
C Shell: `setenv SASORA V7`

- *Make sure that the environment variable `$TWO_TASK` is set to a valid Oracle alias name (i.e. `oracle_alias`) which is defined in the Oracle `tnsnames.ora` file. The following syntax is used to set variables:*

Bourne Shell: `export TWO_TASK=oracle_alias`
C Shell: `setenv TWO_TASK oracle_alias`

- *For Solaris 2 platforms, make sure that the environment variable \$LD_LIBRARY_PATH is set appropriately for your site. The following syntax can be used to set the variables:*

Bourne Shell:

```
LD_LIBRARY_PATH= /lib:/usr/lib:/usr/openwin/lib
export LD_LIBRARY_PATH
```

C Shell:

```
setenv LD_LIBRARY_PATH /lib:/usr/lib:/usr/openwin/lib
```

Once the above requirements have been met, you should be able to successfully connect to the ORACLE database(s) using the following steps:

- 1) *Invoke SAS from the **exact** environment and userid you used when you successfully connected to the database(s) using the ORACLE SQLPLUS utility.*
- 2) *To verify that you are able to successfully connect to an ORACLE database, submit the following PROC SQL Pass-Through code from the SAS Program Editor. Be sure the `usr_name`, `password`, `connect_string` and `tbl_name` are the same values used with the ORACLE SQLPLUS utility.*

```
proc sql;
connect to oracle(user=usr_name orapw=password
                 path="@connect_string");
  select * from connection to oracle
  (select * from tbl_name);
disconnect from oracle;
quit;
```

NOTE: *It is highly recommended to set the ORACLE environment variable, \$TWO_TASK. If \$TWO_TASK is set then the path= option is **not** required. You must either have the path= set correctly or the \$TWO_TASK environment variable set. **This is a SAS requirement.***

The user= and orapw= options can also be omitted, if your ORACLE installation and/or server supports OPS\$sysid and is set up to use the OPS\$sysid support.

- 3) *If, when using SAS/ACCESS, you are unable to connect to the same ORACLE database from the **exact** environment and userid that you were able to successfully connect to using the SQLPLUS utility, then contact SAS UNIX Technical Support at (919) 677-8008 and have the following information available:*

- the exact operating system release level and machine type
- the exact ORACLE release level
- is the ORACLE database local or remote?
- the exact release of SQLNET
- the environment variable settings
- the results from the steps documented above for both "Using the ORACLE SQLPLUS Utility" and "Using SAS/ACCESS."
- the contents of the file `$ORACLE_HOME/rdbms/lib/sysliblist`
- the results from running the *PRO*C Demo* as follows (be sure the environment variable `$ORACLE_HOME` is set first):

```
unix_prompt> cp $ORACLE_HOME/proc/demo/sample1.pc /tmp
unix_prompt> cd /tmp
unix_prompt> make -f $ORACLE_HOME/proc/demo/proc.mk sample1
```

- 4) Once you have connected successfully and want more details on using the ACCESS Procedure, DBLOAD Procedure or the SQL Procedure Pass-Through Facility, reference the SAS manuals listed in the CONCLUSION of this paper.

Conclusion

This paper discussed the various aspects of connecting to ORACLE databases with both the ORACLE SQLPLUS utility outside of SAS and with SAS using the SAS/ACCESS Interface to ORACLE product on UNIX platforms. This discussion includes details and examples of using the SQLPLUS utility and the SAS SQL Procedure Pass-Through Facility.

For detailed information on the ORACLE SQL commands used in this report, please refer to your ORACLE manuals or ask your Database Administrator for assistance.

For detailed information on the SAS procedures PROC ACCESS, PROC DBLOAD, PROC SQL PASS-THROUGH, reference the following manuals:

SAS/ACCESS Software for Relational Databases Reference, Version 6 (# 55144)

SAS/ACCESS Interface to ORACLE Usage and Reference, Version 6 (# 56082)

SAS/ACCESS Software Changes and Enhancements, SQL Procedure Pass-Through Facility, Version 6 (# 55237))

SAS Software Changes and Enhancements, Release 6.11 (# 55300)

*SAS Technical Report, Append Capability for PROC DBLOAD (TS-347)
Getting Started with SAS/ACCESS Software, Version 6 (# 55103)*

SAS/ACCESS INTERFACE TO ORACLE SUPPORT ADDENDUM

6.07 PLATFORMS

<u>Platform</u>	<u>Oracle</u>
-----------------	---------------

1. SUN4	TS203
---------	-------

** SQL passthrough only; shipped experimental and rolled to production in the field.*

2. HPU8	TS210
---------	-------

** SQL passthrough only; shipped experimental and rolled to production in the field.*

3. AIXR	TS105
---------	-------

** SQL passthrough only; shipped experimental and rolled to production in the field.*

6.09 PLATFORMS

<u>Platform</u>	<u>Oracle</u>
-----------------	---------------

1. SUN4	TS027
---------	-------

2. SOL2	TS037
---------	-------

3. HPU8	TS027
---------	-------

4. AIXR	TS027
---------	-------

6.10 PLATFORMS

<u>Platform</u>	<u>Oracle</u>
-----------------	---------------

1. Alpha/OSF	TS018
--------------	-------

** SQL passthrough only; shipped experimental and rolled to production in the field.*

2. MABI	
---------	--

- SGI	TS018
-------	-------

- Pyramid	TS018
-----------	-------

** SQL passthrough only; shipped experimental and will go in production for 6.11 Wave 2.*

3. IABI	
---------	--

- NCR	TS018
-------	-------

- Sequent	TS018
-----------	-------

** SQL passthrough only; shipped experimental and have no plans to go production.*

6.11 PLATFORMS

<u>Platform</u>	<u>Oracle</u>
1. SUN4 & SOL2	TS020
2. HPU8	TS020
3. AIXR	TS020
4. Digital UNIX	TS040 *SQL passthrough, DBLOAD
5. MABI	
- SGI	TS040
- Pyramid	TS040
	*SQL passthrough, DBLOAD for both hosts
6. IABI	
- NCR	N/A
- Sequent	N/A

6.12 BETA PLATFORMS

<u>Platform</u>	<u>Oracle</u>
1. SUN4 & SOL2	TS005
2. HPU8	TS005
3. AIXR	TS005