

TS-518

**SAS/ACCESS® Guidelines for Connecting to Relational
Database Systems 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 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 DB2/6000®, INFORMIX®, INGRES®, ORACLE®, and/or SYBASE® on UNIX platforms.

Introduction

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 product availability support.

With every relational database management systems (DBMS), a utility (i.e., SQLPLUS, ISQL, DB-ACCESS, DB2, SQL) is available to provide users the capability to retrieve and create data stored in the 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, Solaris 2, or AIX® system using SAS 6.11 TS020, and demonstrate the use of the DBMS utilities and successful connections to databases using SAS/ACCESS.

DB2

The SAS/ACCESS Interface to DB2/6000 product only includes support for the SQL Procedure Pass-Through Facility and the SAS DBLOAD Procedure. The ACCESS Procedure is not available for the DB2/6000 interface.

USING THE DB2/6000 DB2 UTILITY

In order to connect to a DB2/6000 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 DB2/6000 Command Line Processor.

*Once a successful database connection is made using the DB2 Command Line Processor, 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 a DB2/6000 database outside of SAS successfully.

- 1) *Invoke the DB2 Command Line Processor as follows:*

```
unix_prompt> db2
```

- 2) *Once you successfully invoke the DB2 Command Line Processor, you will receive information similar to the following. If you are having problems invoking the DB2 Command Line Processor, you will need to contact your DBA or IBM Technical Support for assistance.*

```
(c) Copyright IBM Corporation 1993,1995  
Command Line Processor for DB2 SDK 2.1.1
```

You can issue database manager commands and SQL statements from the command prompt. For example:

```
db2 => connect to sample  
db2 => bind sample.bnd
```

For general help, type: ?.

For command help, type: ? command, where command can be the first few keywords of a database manager command. For example:

```
? CATALOG DATABASE for help on the CATALOG DATABASE command  
? CATALOG           for help on all of the CATALOG commands.
```

To exit db2 interactive mode, type QUIT at the command prompt. Outside interactive mode, all commands must be prefixed with 'db2'. To list the current command option settings, type LIST COMMAND OPTIONS.

For more detailed help, refer to the Online Reference Manual.

```
db2 =>
```

- 3) *From the DB2 command line, connect to the database(s) that you plan to connect to, with SAS, using the system default user name and password (where db_name is a valid database name for connecting):*

```
db2 => connect to db_name
```

Use the following syntax, if you need to specify the user name and password (where db_name, usr_name and password are valid values for the database to which you are connecting):

```
db2 => connect to db_name user usr_name using password
```

- 4) *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 IBM Technical Support for assistance.*

```
Database Connection Information
Database product           = DB2/6000 PE 1.2.0
SQL authorization ID      = SAS
Local database alias      = DB_NAME
```

db2 =>

- 5) *To verify that you have been granted the appropriate privileges, submit some SQL code from the DB2 command line, similar to what you plan to submit from within SAS. If you experience problems with privileges or submitting SQL from the DB2 Command Line Processor, and you will need to contact your DBA or IBM 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):*

db2 => select * from tbl_name

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
A00	SPIFFY COMPUTER SERVICE DIV.	000010	A00	-
D01	DEVELOPMENT CENTER	-	A00	-
E11	OPERATIONS	000090	E01	-
C01	INFORMATION CENTER	000030	A00	-
D11	MANUFACTURING SYSTEMS	000060	D01	-

5 record(s) selected.

- 6) *You have just demonstrated that you can connect to the DB2 database outside of SAS. Before you attempt to bring up SAS and try to connect to the database using SAS/ACCESS, be sure that you use the **exact** usr_name, password, and db_name that you used with the DB2 Command Line Processor.*

Using SAS/ACCESS Interface to DB2/6000

Before you attempt to connect to a DB2/6000 database using SAS/ACCESS, you must be able to connect to the same database outside of SAS using the DB2 Command Line Processor as demonstrated above. Once you have verified that you can connect to the database(s) using the DB2 Command Line Processor, 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 DB2 Command Line Processor and connected to the database(s) successfully.
- You have performed the SAS/ACCESS Interface to DB2/6000 Installation Instructions from APPENDIX B of the "Installation Instructions for the SAS System under UNIX Environments".

Note that, the SAS 6.11 Installation Instructions fail to state in Appendix B on pages 47 - 49 that there is a pre-linked DB2 SAS/ACCESS module shipped with SAS 6.11 and the custom install link is not required. The name of the module is !SASROOT/sasexe/dbi/sasptdb2. Therefore, be sure that the !SASROOT/sasexe/dbi/sasptdb2 module exists and it has read and execute permissions.

- Make sure that the DB2 environment variable \$INSTHOME is set to the location of the DB2/6000 home/root directory .
- Make sure that the DB2 environment variable is set \$DB2INSTANCE.
- Make sure that the environment variable \$LIBPATH is set to the location of the DB2 home directory. SAS will use the libdb2.a "shared" library in that location. The following syntax can be used to set the variables:

Bourne Shell: LIBPATH=\$INSTHOME/lib:\$LIBPATH
export LIBPATH

C Shell: setenv LIBPATH \$INSTHOME/lib:\$LIBPATH

- You must bind the SAS application package DB2.SASDB2 to all databases with which you intend to use the SAS/ACCESS interface. The bind should be performed with the following DB2 command from within the DB2 Command Line Processor:

```
db2 => BIND !SASROOT/sas611/saspgm/dbi/obj/sasdb2.bnd
```

- Finally, you should grant execute privileges on the DB2.SASDB2 package to your users using the DB2 Command Line Processor. For example,

```
db2 => GRANT execute on package DB2.SASDB2 to public
```

Once the above requirements have been met, you should be able to successfully connect to the DB2/6000 database(s) using the following steps:

- 1) *Invoke SAS with the **exact** environment and userid you used when you successfully connected to the database(s) using the DB2 Command Line Processor.*
- 2) *To verify that you are able to successfully connect to a DB2/6000 database, submit the following PROC SQL Pass-Through code from the SAS Program Editor. Be sure the db_name, usr_name, password and tbl_name are the same values used with the DB2/6000 DB2 Command Line Processor.*

```
proc sql;  
connect to db2(db=db_name user=usr_name using=password);  
  select * from connection to db2  
    (select * from tbl_name);  
disconnect from db2;  
quit;
```

NOTE: If the usr_name and password are the same as the operating system user name and password then the user= and using= options are not required. (They are only required if they were required when you connected using the DB2 Command Line Processor.)

- 3) *If, when using SAS/ACCESS, you are unable to connect to the same DB2 database from the **exact** environment and userid that you were able to successfully connect to using the DB2 Command Line Processor, 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 DB2 release level*
 - *is the DB2 database local or remote. If remote, is the Client Application Enabler (CAE) used? If so, what is the CAE release?*
 - *the environment variable settings*
 - *the results from the steps documented above for both "Using the DB2/6000 DB2 utility" and "Using SAS/ACCESS."*
- 4) *Once you have connected successfully and want more details on using the DBLOAD Procedure or the SQL Procedure Pass-Through Facility, reference the SAS manuals listed in the CONCLUSION of this paper.*

INFORMIX

The SAS/ACCESS Interface to INFORMIX product only includes support for the SQL Procedure Pass-Through Facility. The ACCESS and DBLOAD Procedures are not available for the INFORMIX interface.

Using the INFORMIX DB-Access Utility

In order to connect to an INFORMIX 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 INFORMIX DB-Access utility.

*Once a successful database connection is made using DB-Access, you should be able to connect to the same database(s) from within SAS with the **same** environment and userid. Below are some simple steps to verify whether you are able to connect to an INFORMIX database outside of SAS successfully.*

- 1) *Invoke the INFORMIX DB-Access menu system:*

```
unix_prompt> db-access
```

- 2) *Once you successfully invoke the INFORMIX DB-Access menu system, you will receive a menu similar to the following. If you are having problems invoking the DB-Access utility, you will need to contact your DBA or INFORMIX Technical Support for assistance.*

```
DB-ACCESS:  Query-language Connection Database Table Session Exit  
Use SQL query language.  
----- Press CTRL-W for Help -----
```

- 3) *From the DB-Access MAIN menu, space over to the CONNECTION option and press return.*
- 4) *From the CONNECTION submenu, space over to the CONNECT option and press return.*
- 5) *You must select a database server listed under the submenu, SELECT DATABASE SERVER, by using the arrow keys to move to the database of your choice and pressing return.*
- 6) *Select a database listed under the submenu, SELECT DATABASE, by using the arrow keys and pressing return.*

- 7) *Now EXIT the CONNECTION submenu, by spacing over to the EXIT option and pressing return. This will return back to the MAIN DB-Access menu.*
- 8) *Select the QUERY-LANGUAGE option from the MAIN menu using the spacebar and pressing return.*
- 9) *To verify that you have been granted the appropriate privileges, submit an SQL query from DB-Access, similar to what you plan to submit from within SAS. If you experience problems with privileges, submitting SQL or any of the menu selections from DB-Access, you will need to contact your DBA or INFORMIX Technical Support for assistance.*

From the SQL submenu, select the NEW option and enter a valid SQL query similar to the following: (where tbl_name is a valid table name in the database that you are connected):

```
select * from tbl_name;
```

This should return data from the table that satisfies your query. If not, contact your DBA or INFORMIX Technical Support for assistance.

- 10) *You have just demonstrated that you can connect to an INFORMIX database outside of SAS. Before you attempt to bring up SAS and try to connect using the SAS/ACCESS SQL Procedure Pass-Through Facility, be sure to use the **exact** database and table names that you used with the INFORMIX DB-Access utility.*

Using SAS/ACCESS Interface to INFORMIX

Before you attempt to connect to an INFORMIX database using SAS/ACCESS, you must be able to connect to the same database outside of SAS using the INFORMIX DB-Access utility as demonstrated above. Once you have verified that you can connect to the database(s) using DB-Access, 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 that you used to invoke DB-Access and connected to the database(s) successfully.*
- *Make sure that the !SASROOT/sasexe/dbi/sasnfx module, along with all other INFORMIX 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.*

- You have successfully performed the SAS/ACCESS Interface to INFORMIX Installation Instructions from APPENDIX B of the "Installation Instructions for the SAS System under UNIX Environments".
- Make sure that the environment variable `$INFORMIXDIR` is set to the location of the INFORMIX home/root directory.

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

- 1) Invoke SAS with the **exact** environment and userid you used when you successfully connected to the database(s) using the DB-Access utility.
- 2) To verify your ability to successfully connect to an INFORMIX database, submit the following PROC SQL Pass-Through code from the SAS Program Editor. Be sure the `db_name` and `tbl_name` are the same values used with the INFORMIX DB-Access utility.

```
proc sql;
connect to informix(database=db_name);
  select * from connection to informix
    (select * from tbl_name);
disconnect from informix;
quit;
```

NOTE: If the INFORMIX environment variable, `$DBDATABASE`, is set to the database you want to connect to, then the `database=` option is not required. If this environment variable is set, you can also omit the `connect` statement because an implicit connection is performed when the `select` or `execute` statement is passed to INFORMIX.

- 3) If, when using SAS/ACCESS, you are unable to connect to the same INFORMIX database from the **exact** environment and userid that you were able to successfully connect to using the INFORMIX DB-Access 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 INFORMIX release level and whether it is INFORMIX-SE or INFORMIX-ONLINE
 - is the INFORMIX database local or remote?
 - the environment variable settings
 - the results from the steps documented above for both "Using the INFORMIX DB-Access Utility" and "Using SAS/ACCESS."

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

INGRES

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

USING THE INGRES SQL UTILITY

In order to be able to connect to an INGRES 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 INGRES SQL utility.

*Once a successful database connection is made using the SQL 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 INGRES database outside of SAS successfully.*

- 1) *Invoke the SQL utility as follows (where db_name is a valid database name for connecting):*

```
unix_prompt> sql db_name
```

- 2) *Once you successfully invoke the SQL utility and connect to the database specified, you will receive a prompt similar to the following. If you are having problems invoking the INGRES SQL utility or connecting to the database, you will need to contact your DBA or INGRES Technical Support for assistance.*

```
INGRES TERMINAL MONITOR Copyright (c) 1981, 1991 Computer  
Associates Intl, Inc.  
00OpenINGRES SPARC SOLARIS Version OpING 1.1/04 (su4.us5/00)  
login Tue Aug 20 10:59:06 1996
```

```
continue  
*
```

- 3) *To verify that you have been granted the appropriate privileges, submit an SQL query from INGRES, similar to what you plan to submit from within SAS. If you experience problems with privileges or submitting SQL from the INGRES SQL utility, this is an INGRES issue and you will need to contact your DBA or INGRES 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 you just connected to):

```
* select * from tbl_name \g
```

Executing . . .

name	salary	dept	div	mgr	birthdate	num
Aitken	\$50000.00	sales	NE	Verducci	12-mar-1947	2
Blumberg	\$32000.00	sales	SC	Clark	06-dec-1952	2
Brodie	\$40000.00	sales	NC	Clark	01-feb-1948	2
Cameron	\$35000.00	tech sup	SE	Shigio	15-aug-1960	2

- 4) *You have just demonstrated that you can connect to an INGRES 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** db_name and tbl_name that you used with the INGRES SQL utility.*

Using SAS/ACCESS Interface to INGRES

Before you attempt to connect to an INGRES database using SAS/ACCESS, you must be able to connect to the same database outside of SAS using the INGRES SQL utility as demonstrated above. Once you have verified that you can connect to the database(s) using the SQL 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 that you used to invoke the INGRES SQL utility and connected to the database(s) successfully.*
- *Make sure that the !SASROOT/sasexe/dbi/sasing module, along with all other INGRES 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.*
- *You have successfully performed the SAS/ACCESS Interface to INGRES Installation Instructions from APPENDIX B of the "Installation Instructions for the SAS System under UNIX Environments".*
- *Make sure that the environment variable \$II_SYSTEM is set to the location of the INGRES home/root directory.*

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

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

```
proc sql;
connect to ingres(database=db_name);
  select * from connection to ingres
    (select * from tbl_name);
disconnect from ingres;
quit;
```

- 3) If, when using SAS/ACCESS, you are unable to connect to the same INGRES database from the **exact** environment and userid that you were able to successfully connect to using the INGRES SQL 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 INGRES release level
 - is the INGRES database local or remote?
 - the environment variable settings
 - the results from the steps documented above for both "Using the INGRES SQL utility" and "Using SAS/ACCESS."
- 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.

ORACLE

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

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 the database for which you are connecting. If you do not know what these values are or should be, or you 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
```

- 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 that you 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 that 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 with 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.*

SYBASE

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

Using the SYBASE ISQL UTILITY

In order to connect to a SYBASE 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 SYBASE ISQL utility.

*Once a successful database connection is made using the ISQL 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 a SYBASE database outside of SAS successfully.

- 1) *Invoke the SYBASE ISQL utility as follows, where `usr_name`, and `svr_name` are valid values for the database for which you are connecting. 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 SYBASE Technical Support for assistance. If you are not able to connect to SYBASE using the ISQL utility using the **exact** syntax demonstrated below, you will not be able to use SAS/ACCESS to connect to the database either.*

```
unix_prompt> isql -S svr_name -U usr_name
```

- 2) *You will receive the following prompt for a password and you must enter a valid password. If you have problems with your password, contact you DBA for assistance.*

```
Password: password
```

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

```
1>
```

- 3) *To verify that you have been granted the appropriate privileges, submit an SQL query from within ISQL, similar to what you plan to submit from within SAS, If you experience problems with privileges or submitting SQL from ISQL, you will need to contact your DBA or SYBASE 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 you are connected):*

```
1> select * from tbl_name
2> go
```

EMPID	SALARY
-----	-----
123456	50000.00
789012	75000.00
555555	12000.00

```
(3 rows affected)
```

- 4) *You have just demonstrated that you can connect to the SYBASE 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, svr_name and tbl_name that you used with the SYBASE ISQL utility.*

Using SAS/ACCESS Interface to SYBASE

Before you attempt to connect to a SYBASE database using SAS/ACCESS, you must be able to connect to the same database outside of SAS using the SYBASE ISQL utility as demonstrated above. Once you have verified that you can connect to the database(s) using the ISQL 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 from the **exact** environment and userid that you used to invoke the ISQL utility and connected the database(s) successfully.*
- *You have successfully performed the SAS/ACCESS Interface to SYBASE Installation Instructions from APPENDIX B of the "Installation Instructions for the SAS System under UNIX Environments".*
- *Make sure that the !SASROOT/sasexe/dbi/sassyb module, along with all other SYBASE 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.*
- *For Solaris 2 platforms, make sure that you add the location of the SYBASE shared libraries to your systems \$LD_LIBRARY_PATH environment variable before executing the SAS/ACCESS Interface to SYBASE and SQL Server. The following syntax can be used to set the variables:*

Bourne Shell:

```
LD_LIBRARY_PATH= $SYBASE/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

C Shell:

```
setenv LD_LIBRARY_PATH $LD_LIBRARY_PATH:$SYBASE/lib
```

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

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

```
proc sql;  
connect to sybase(user="usr_name" password="password"  
                 database="db_name" server="svr_name");  
    select * from connection to sybase  
        (select * from "tbl_name");  
disconnect from sybase;  
quit;
```

NOTE: SYBASE and the SQL Server are very case sensitive, therefore, it is highly recommended that you include the `usr_name`, `password`, `svr_name`, `db_name` and `tbl_name` in quotes (using the exact spelling, uppercase and lowercase letters used with the ISQL utility). Therefore, SAS will not uppercase the values but just pass it to SYBASE.

- 3) *If, when using SAS/ACCESS, you are unable to connect to the same SYBASE database from the **exact** environment and userid that you were able to successfully connect to using the ISQL 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 SYBASE release level*
 - *is the SYBASE database local or remote?*
 - *the environment variable settings*
 - *the results from the steps documented above for both "Using the SYBASE ISQL Utility" and "Using SAS/ACCESS."*
- 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 DBMS databases with both the DBMS utility outside of SAS or with SAS using the SAS/ACCESS product on UNIX platforms. This discussion includes details and examples using the DBMS utility and the SAS SQL Procedure Pass-Through Facility.

For detailed information on the DBMS commands used in this report, please refer to your DBMS 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 Interface to SYBASE and SQL Server Usage and Reference, Version 6 (# 56081)

SAS/ACCESS Interface to INGRES Usage and Reference, Version 6 (# 56072)

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

SAS/ACCESS Interface to INFORMIX Technical Report, SQL Procedure Pass-Through Facility, Release 6.09 (# 55119)

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 FOR UNIX PRODUCT AVAILABILITY ADDENDUM

6.07 PLATFORMS

<u>Platform</u>	<u>Oracle</u>	<u>Sybase</u>
1. SUN4	TS203	TS203

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

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

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

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

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

6.09 PLATFORMS

<u>Platform</u>	<u>Oracle</u>	<u>Sybase</u>	<u>Ingres</u>	<u>Informix</u>	<u>DB2/6000</u>
1. SUN4	TS027	TS027	TS027	TS037	N/A
				*passthru	
2. SOL2	TS037	TS037	TS037	TS037	N/A
				*passthru	
3. HPU8	TS027	TS027	TS027	TS037	N/A
				*passthru	
4. AIXR	TS027	TS027	TS027	TS037	TS042
				*passthru	*passthru

6.10 PLATFORMS

<u>Platform</u>	<u>Oracle</u>	<u>Sybase</u>
1. Alpha/OSF	TS018	N/A

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

2. MABI

- SGI	TS018	TS018
- Pyramid	TS018	TS018

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

3. IABI

- NCR	TS018	TS018
- Sequent	TS018	TS018

* passthru only; shipped experimental and have no plans to go production as of yet.

6.11 PLATFORMS

<u>Platform</u>	<u>Oracle</u>	<u>Sybase</u>	<u>Ingres</u>	<u>Informix</u>	<u>DB2/6000</u>	<u>PCFF</u>
1. SUN4 & SOL2	TS020	TS020	TS020	TS020 *passthru	N/A	TS020 *PROC DBF,DIF
2. HPU8	TS020	TS020	TS020	TS020 *passthru	N/A	TS020 *PROC DBF,DIF
3. AIXR	TS020	TS020	TS020	TS020 *passthru	TS020 *passthru *DBLOAD	TS020 *PROC DBF,DIF
4. Digital UNIX	TS040 *passthru,DBLOAD	N/A	TS040 *passthru,DBLOAD	N/A	N/A	TS040 *PROC DBF,DIF
5. MABI						
- SGI	TS040	N/A	N/A	N/A	N/A	TS040
- Pyramid	TS040 *passthru,DBLOAD	N/A	N/A	N/A	N/A	TS040 *PROC DBF,DIF
*passthru,DBLOAD for both hosts						
6. IABI						
- NCR	N/A	N/A	N/A	N/A	N/A	TS040
- Sequent	N/A	N/A	N/A	N/A	N/A	TS040 *PROC DBF,DIF

6.12 BETA PLATFORMS

<u>Platform</u>	<u>Oracle</u>	<u>Sybase</u>	<u>Ingres</u>	<u>Informix</u>	<u>DB2/6000</u>	<u>ODBC</u>	<u>PCFF</u>
1. SUN4 & SOL2	TS005	TS005	TS005	TS005 *passthru	N/A	N/A	TS005 *PROC DBF,DIF
2. HPU8	TS005	TS005	TS005	TS005 *passthru	N/A	TS005	TS005 *PROC DBF,DIF
3. AIXR	TS005	TS005	TS005	TS005 *passthru	TS005 *passthru *DBLOAD	N/A	TS005 *PROC DBF,DIF