

Accessing Data from Your PC Using Version 7 of the SAS® System

Roger E. Sanders, SAS Institute Inc., Cary, NC

ABSTRACT

Data access technology is changing rapidly in today's computing environment, especially on the PC platform. The Version 7 release of the SAS System is the first release that contains a newly designed suite of SAS/ACCESS® software products that strive to take advantage of recent technological advances in data access.

This paper provides an overview of the SAS external data access capabilities that are available on the PC. Emphasis is placed on the SAS/ACCESS products, the Import/Export Facility, ODBC drivers that are available, and conversion utilities for existing MS SQL Server and AS/400® view descriptors.

INTRODUCTION

With the Version 7 release of the SAS System, SAS/ACCESS software and the Import/Export Facility contain many important enhancements that allow SAS users to transparently integrate their Database Management Systems with their SAS business solutions.

The new DYNAMIC LIBNAME ENGINES enable the SAS user to issue a LIBNAME statement to connect to a relational DBMS server and read, update and create relational DBMS tables directly. In fact, with relational DBMSs no static Access or View descriptors are required. These engines are available for ORACLE®, SYBASE®, ODBC, and DB2® for Common Servers. Additional features and performance improvements include support for table and column names up to 32 bytes, case-sensitive DBMS object names, line mode updating of Descriptors using PROC ACCESS, passing joins directly to the DBMS, integration of PROC SQL views, concurrency control, and more. PROC ACCESS, PROC DBLOAD, SQL Pass-through, and VIEW Descriptors are also supported for Version 6.x compatibility.

Two new procedures, PROC IMPORT and PROC EXPORT have been added to the Import/Export Facility to enable the SAS user to issue import and/or exports from a SAS program. Two new data sources, Microsoft Access and Excel 97, have also been added.

A conversion utility has also been developed to aid in the conversion of existing AS/400 and MS SQL view descriptors to ODBC view descriptors.

SAS/ACCESS SOFTWARE

SAS/ACCESS software is a set of products for the SAS System that provide a way to access data that is stored in other database management systems (DBMS) and several PC file formats. SAS/ACCESS products allow you to transparently interact with data stored in any non SAS data source for which a SAS/ACCESS interface (engine) is available. Each SAS/ACCESS engine provides one or more of the following capabilities:

- The ability to read data from an external data source.
- The ability to create a SAS view or data set from data stored in an external data source.
- The ability to update data stored in an external data source.
- The ability to write data stored in a SAS data set to an external data source.
- The ability to pass SQL logic directly to a non SAS DBMS.

In Release 6.12 of the SAS System, you could use either PROC ACCESS or the ACCESS Window to create an access descriptor that contained essential information about the structure of a non SAS data source. Using this access descriptor, one or more view descriptors could be created that defined some or all of the metadata that the access descriptor described. Once created, view descriptors could be processed in the same manner as SAS data sets.

In Version 7 of the SAS System, all SAS/ACCESS engines that communicate with relational databases (e.g. Oracle, Sybase, ODBC, and DB2 for Common Servers) have been completely redesigned and rewritten. Depending upon user needs and the capabilities of the SAS/ACCESS engine being used, one or more of the following Version 6 features are supported:

- Version 6 access and view descriptor compatibility
Version 7 SAS/ACCESS engines can read and update all existing Version 6.x access and view descriptors. Likewise, all existing SAS/ACCESS procedure line mode statements and batch SAS jobs will continue to run in Version 7.
- Version 6 PROC SQL Pass-Through compatibility
Version 7 SAS/ACCESS engines will support existing Version 6.x PROC SQL Pass-Through views and batch jobs that contain Version 6.x SQL Pass-Through statements will continue to run.

In Version 7 of the SAS System, the ACCESS Window, along with the full-screen interface to licensed SAS/ACCESS products, is no longer available. Because this interface has been removed, a new statement (the UPDATE statement) has been added to all SAS/ACCESS engines so that the contents of existing descriptors can be modified.

RELATIONAL DBMS SAS/ACCESS ENGINES ARE NOW DYNAMIC LIBNAME ENGINES

With Version 7 of the SAS System there is no longer a need to create Access and View Descriptors to access and manipulate data stored in an external relational Database Management System (DBMS). Now you can issue a libname statement with all the required options to establish a successful DBMS connection. Once the libname is assigned successfully, the tables and views in the DBMS can be accessed by using the libref as if it were a SAS data set. For example:

```
LIBNAME DBMS ORACLE USER=SCOTT  
PASSWORD=TIGER PATH=DBIPC1_ORCL;
```

In the example above, "DBMS" is the libref. "ORACLE" is the SAS/ACCESS to ORACLE engine name. "SCOTT" and "TIGER" are the user ID and password that are passed to the Oracle engine to establish a connection. "DBIPC1_ORCL" is the service name that identifies where the Oracle database resides. The USER, PASSWORD, and PATH options provide the information needed to successfully connect to the Oracle DBMS. By default, a connection to the DBMS is established when the libname statement is issued. This connection can then be used to get to the DBMS's tables and views. For example:

```
PROC DATASETS LIB=DBMS; RUN;
```

The above example, when executed, will list all the Oracle database tables/views that are accessible to the Oracle user SCOTT¹:

-----Directory-----

Libref: DBMS
 Engine: ORACLE
 Physical Name: dbipc1_orcl
 Schema/User: scott

#	Name	Memtype
1	BONUS DATA	
2	DEPT	DATA
3	EMP	DATA
4	EMPTBL DATA	
5	SALGRADE	DATA

¹The actual list of tables and views that will appear is dependent upon the DBMS engine that is being used – in some cases all tables and views found in the database will be listed, in other cases only the tables and views owned by the current user are listed and are accessible.

Data stored in any of the accessible tables and views in the DBMS can be viewed by using the libref and one of its members in a PROC PRINT statement. For example:

```
PROC PRINT DATA=DBMS.EMPTBL; RUN;
```

In the above example, EMPTBL is the name of table in an Oracle database that the libref “DBMS” is connected to. When this statement is executed, the rows of data that are stored in the EMPTBL table will be displayed in the Output window:

Obs	EMPNO	ENAME	SAL	HIREDATE
1	7369	SMITH	800.00	17DEC1980:00:00:00
2	7499	ALLEN	1600.00	20FEB1981:00:00:00
3	7521	WARD	1250.00	22FEB1981:00:00:00
4	7566	JONES	2975.00	02APR1981:00:00:00
5	7654	MARTIN	1250.00	28SEP1981:00:00:00
6	7698	BLAKE	2850.00	01MAY1981:00:00:00
7	7782	CLARK	2450.00	09JUN1981:00:00:00
8	7788	SCOTT	3000.00	19APR1987:00:00:00
9	7839	KING	5000.00	17NOV1981:00:00:00
10	7844	TURNER	1500.00	08SEP1981:00:00:00
11	7876	ADAMS	1100.00	23MAY1987:00:00:00
12	7900	JAMES	950.00	03DEC1981:00:00:00
13	7902	FORD	3000.00	03DEC1981:00:00:00
14	7934	MILLER	1300.00	23JAN1982:00:00:00

You can also use the Oracle table EMPTBL in the SAS language as if it were a SAS data set by referencing it as “DBMS.EMPTBL”. See the following examples.

```
PROC CONTENTS DATA=DBMS.EMPTBL; RUN;
```

The output from the previous example looks something like this.

The CONTENTS Procedure

Data Set Name: DBMS.EMPTBL Observations:
 Member Type: DATA Variables: 4
 Engine: ORACLE Indexes: 0
 Created: . Observation Length: 0
 Last Modified: . Deleted Observations: 0
 Protection: Compressed: NO
 Data Set Type: Sorted: NO
 Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Format	Informat	Label
1	EMPNO	Num	8	0	5.	5.	EMPNO
2	ENAME	Char	10	8	\$10.	\$10.	ENAME
4	HIREDATE	Num	8	32	DATETIME20.	DATETIME20.	HIREDATE
3	SAL	Num	8	24	9.2	9.2	SAL

MISCELLANEOUS NEW FEATURES IN VERSION 7 OF SAS/ACCESS SOFTWARE

In addition to supporting many existing Version 6 features, the following enhancements have been added to all SAS/ACCESS engines that communicate with relational databases in Version 7 of the SAS System:

LINE MODE UPDATING OF ACCESS/VIEW DESCRIPTORS

Version 7 PROC ACCESS allows line mode updating of Access and View descriptors. The UPDATE statement in PROC ACCESS is provided for quick updates, such as re-specifying the password or DBMS server name that may have changed since the Access/View descriptors were created. For example:

```
PROC ACCESS DBMS=ORACLE;  

UPDATE SASUSER.OLDVD.VIEW;  

PATH="NEW_SERVER";  

RUN;
```

The above PROC ACCESS code modifies an existing View Descriptor by assigning a new value for the PATH option.

LONG NAME SUPPORT

Version 7 SAS/ACCESS engines now support 32 byte member and variable names which means that in most cases the table and column names used in the DBMS are the same names used in SAS – truncation no longer occurs. Also, mixed case member and variable names are now supported.

CHARACTER VARIABLES UP TO 32K BYTES

Version 7 SAS allows character variables of size up to 32k. Therefore, SAS/ACCESS engines can now retrieve up to 32k bytes from DBMS columns into SAS variables. Prior to Version 7, the limit was 200 bytes.

MACRO VARIABLES SET BY SAS/ACCESS SOFTWARE

SAS/ACCESS software now has SYSDBRC and SYSDBMSG macro variables that provide the status of the last SAS/ACCESS statement executed. SYSDBRC is 0 if the statement is completed successfully. If an error was encountered, SYSDBRC has the error code and SYSDBMSG has the error message. In Version 6.x SAS/ACCESS, this functionality was provided only in the SQL Pass-through environment with SQLXRC and SQLXMSG macro variables.

BETTER USE OF INDEXES

Version 7 SAS/ACCESS engine implementation techniques now make better use of DBMS table indexes in DBMS queries. For example:

```
LIBNAME DBMS ORACLE USER=SCOTT
  PASSWORD=TIGER PATH=DBIPC1_ORCL;

DATA KEYVALUES;
  DEPTNO=30; OUTPUT;
  DEPTNO=10; OUTPUT;
RUN;

PROC SQL;
  SELECT EMPLNAME FROM
    DBMS.DEPT (DBINDEX=YES), KEYVALUES
  WHERE DEPT.DEPTNO = KEYVALUES.DEPTNO;
RUN;
```

In the example above, the DEPTNO index for the table DEPT in the Oracle database that the libref "DBMS" refers to is used to query the DEPT table using values from the KEYVALUES data set. In this case the join between a SAS data set and a DBMS table is optimized by the DBMS index.

PASSING JOINS TO RELATIONAL DATABASES

When PROC SQL statements that contain joins are used in conjunction with a Version 7 SAS/ACCESS engine, the join is passed to the DBMS for processing.

OUTPUT SUPPORT

In Version 6 of the SAS System, the only way to create DBMS tables with a SAS/ACCESS product was to use PROC DBLOAD. In Version 7 of the SAS System, DBMS engines can create DBMS tables just like the SAS base engine can create SAS data sets -- by using either the DATA step or by using any of the PROCs like PROC COPY that result in the creation of a SAS data set. This is an added feature of Dynamic Libnames. For example:

```
LIBNAME DBMS ORACLE USER=SCOTT
  PASSWORD=TIGER PATH=DBIPC1_ORCL;

DATA DBMS.PAYROLL;
  INPUT EMPNAME $30. PAYCHECK 8.2;
  INFILE 'PAYSTUBS.TXT';
RUN;
```

In the example above, the table PAYROLL is created in the Oracle database that the libref "DBMS" refers to and then it is loaded with the data stored in the file PAYSTUBS.TXT.

DIRECTORY SERVICES

Version 7 SAS/ACCESS engines can now list and drop DBMS tables through PROC SQL and PROC DATASETS. Refer to the PROC DATASETS example above.

SAS/ACCESS TO PC FILE FORMATS

In Version 6.12 of the SAS System, the SAS/ACCESS to PC File Formats product contained engines that communicated with several PC file formats such as Lotus, Excel, and dBASE. The specific engine to use was determined with the DBMS= option of the ACCESS and DBLOAD procedures. The value of this option usually corresponded with the extension used by the application that created the file (i.e. DBF [dBASE], DIF [Data Interchange Format], WK1, WK3, WK4 [Lotus versions 1, 3, and 4], and XLS [Excel versions 4.0 to 7.0]).

In Version 7 of the SAS System, all SAS/ACCESS engines that communicate with PC File Formats essentially remain the same:

- Variable names in view descriptors are limited to 8 characters
- Character values are limited to 200 bytes
- Functionality of Version 7 descriptors and Version 6 descriptors is the same
- Functionality of Version 7 PROC DBLOAD and Version 6 PROC DBLOAD is the same

Two major differences between the Version 6 and the Version 7 SAS/ACCESS to PC File Formats product:

- Full-screen support for access and view descriptor creation is no longer available in Version 7.
- The same line-mode UPDATEing of Access and View descriptors that has been added to the SAS/ACCESS products for relational DBMSs has been added to SAS/ACCESS to PC File Formats.

CHANGES IN IMPORT/EXPORT

The Import/Export facility is a tool in the SAS System that can be used to transfer data between external data sources and SAS data sets. In Release 6.12 of the SAS System, you could use the Import/Export tool to transfer data between the SAS System and one or more of the following external data file types¹:

- dBASE files - (*.DBF)
- Lotus 1-2-3 files - (*.WK1), (*.WK3), and (*.WK4)
- Excel Version 4 files - (*.XLS)
- Excel Version 5 and 7 files - (*.XLS)
- ASCII Delimited files files - (*.*)
- Comma Delimited files - (*.CSV)
- Tab Delimited files - (*.TXT)

¹The actual external data file types supported depend upon the operating system the SAS System is installed on and whether or not the SAS/ACCESS to PC File Formats product is licensed.

In Release 6.12 of the SAS System, the Import and Export menu items on the File pull-down menu are used to invoke the Import/Export facility. When either menu option is selected the Import or Export wizard is displayed and you simply follow the directions to complete the process. When all the necessary information is entered on each of the wizard's screens, data stored in an external file is read into a SAS data set (for Import) or data stored in a SAS data set is written to an external file (for Export). If the external file specified is a delimited text file, the EFI facility is used to transfer the data. If the external file that is specified is a Lotus 1-2-3 or an Excel file, the SAS/ACCESS to PC File Formats product is invoked behind the scenes.

All features that the Import/Export facility had in Version 6 are available in Version 7 of the SAS System. Additionally, the Import/Export facility can now be accessed by two new procedures: PROC IMPORT and PROC EXPORT. Also, in Version 7 of the SAS System, the Import/Export facility will be able to transfer data between the SAS System and the following new external data file types²:

- Excel 97 files - (*.XLS)
- Microsoft Access databases - (*.MDB)

²Provided the SAS System is installed on either Windows 95 or Windows NT and the SAS/ACCESS to PC File Formats product is licensed.

In Release 6.12 of the SAS System, a side effect of the Import/Export facility existed because the SAS/ACCESS to PC File Formats product was used to transfer data between SAS data sets and Lotus 1-2-3 and Excel files. This side effect allowed a user to load SAS statements used by the Import/Export facility into the Program Editor window with the RECALL command after an Import or Export operation completed. In

Version 7 of the SAS System, PROC ACCESS and PROC DBLOAD are still used for Lotus 1-2-3 and some versions of Excel file formats, however the SAS statements generated behind the scenes can no longer be RECALLED. This change occurred because of internal architectural changes within the SAS System. One of the goals of the Import/Export facility is to provide users with the capability to transfer data between the SAS System and an external data file while shielding them from the underlying implementation. By using the IMPORT and EXPORT procedures, users should be able to achieve the same results without having to understand how the data exchange is taking place.

Since the IMPORT and EXPORT procedures have been added, the wizard has been modified to take advantage of them. A new panel has been added to the wizard that allows you to generate and store PROC IMPORT and/or PROC EXPORT procedure statements that are based on the selections made in the wizard in a SAS program file. This file can be loaded into the Program Editor window and executed later to reproduce the Import/Export process.

ODBC CONVERSION UTILITY

In Release 6.12 of the SAS System, there were SAS/ACCESS engines available that could communicate with AS/400 and Microsoft SQL Server data sources. The SAS/ACCESS engine for AS/400 was a 16-bit engine that was available only on OS/2[®]. The SAS/ACCESS engine for MS SQL Server was based on the combined technology of Microsoft and Sybase. Over the last couple of years, IBM[®] has focused on other operating systems in addition to OS/2 and Microsoft and Sybase have terminated their coordinated relational database development efforts. Because of these changes, the native SAS/ACCESS engines for AS/400 and MS SQL Server will not be available in Version 7 of the SAS System. Users needing to access data in either of these two data sources will now have to use the SAS/ACCESS Interface to ODBC. This raises the question "How do I use AS/400 and MS SQL Server view descriptors that I created in Release 6.12 in Version 7?". To answer this question, a conversion program was developed to help users of the SAS/ACCESS Interface to IBM AS/400 and SAS/ACCESS Interface to Sybase and SQL Server products to move from a native DBMS interface connection to an ODBC interface connection (on Windows platforms). The conversion utility is invoked through a SAS procedure and its syntax is as follows:

```
PROC CV2ODBC DBMS = AS400 | MSSQL;
FROM <VIEW> = name_of_v6_created_view <(PW =
password)>;
TO <VIEW> = name_of_new_v7_view;
<DSN = data_source_name_for_ODBC_connection>;
<UID | USER = user_name_for_DBMS_connection>;
<PWD | PASSWORD | PW | PASS = password_for_DBMS
_connection>;
<SAVE <AS> = external_file_name>;
<SUBMIT <STATEMENTS>>;
<REPLACE <ALL> | VIEW | FILE>;
```

- The PW= option for the FROM <VIEW> = statement is needed if the view descriptor referenced in the FROM <VIEW> = statement was created with SAS password security. If a password is specified with the PW= option, the SAS password value supplied will be applied to the Version 7 view descriptor being created.
- The DSN= statement is used to specify the data source name created for the selected DBMS (AS/400 or MS SQL Server). You must register the data source with ODBC (with the ODBC Administrator) before it can be used in the DSN= statement. This statement is optional; if a data source name is not specified, the value assigned to special environment variables is used. For AS/400, the environment variable AS400DSN, by default, contains the value DSN="AS400". For MS SQL Server the environment variable MSSQLDSN, by default, contains the value DSN={name of the ODBC driver specified with the DRIVER= environment option}.

- For AS/400, you can ignore the UID= statement and the PWD= statement if you already have an AS/400 connection established for your PC. Otherwise, you can specify these statements to establish an AS/400 connection. Both statements are optional.

- For MS SQL Server, you must specify the UID= and PWD= statements if the view descriptor referenced in the FROM VIEW= statement was created with a user ID and password. PROC CV2ODBC will validate the values specified with the UID= and PWD= statements and use them to establish the ODBC to DBMS connection. If you do not know or remember the user ID and password values given for the FROM VIEW= view descriptor, you can specify any valid user ID and/or password value(s) and PROC CV2ODBC will use them to establish the ODBC to DBMS connection.

- The SAVE AS= statement allows you to save the generated PROC SQL statements in a file. You may check and/or modify the generated statements as needed and, if necessary, re-submit them on any platform where the SAS/ACCESS ODBC Interface is supported and the appropriate ODBC drivers and DSNs are available. If the SAVE AS= statement is used, PROC CV2ODBC will not automatically submit the generated statements unless you also provide the SUBMIT statement; if the SAVE AS= statement is not used, the SUBMIT statement does not have to be specified.

- If the view descriptor specified in the TO <VIEW> = statement already exists, you can supply the REPLACE VIEW statement to override the existing view descriptor. Existing views are not overridden if the REPLACE VIEW statement is not used.

- If the filename specified in the SAVE AS= statement already exists, you can supply the REPLACE FILE statement to override the existing file. If the filename specified already exists and the REPLACE FILE statement is not specified, all statements generated are appended to the existing file. Existing files are not overridden if the REPLACE FILE statement is not used.

- The REPLACE or REPLACE ALL statement turns on both the REPLACE FILE and the REPLACE VIEW options. Existing views and/or files are not overridden if the REPLACE or the REPLACE ALL statement is not used.

The PROC SQL Pass-Through facility has also been modified so that views that were created with the Version 6 PROC SQL Pass-Through facility will continue to execute -- using ODBC. The conversion process is automatic. In fact, you may not notice that a conversion process is being executed; however, you do have to make sure that the ODBC interface and DBMS client is available and ready to establish a connection. You can do this by performing the following:

For IBM AS/400, you are required to create a data source name. You can do this through the ODBC Administrator. When creating the data source name, you can use the default name ("AS400") that is assigned to the SAS environment variable AS400DSN and subsequently used by the conversion procedure, or you can specify another. You also need to set the environment variable AS400DSN to the created 'data_source_name'. Quotes are required if blank(s) or special characters are included.

For MS SQL Server, if you are using Microsoft's ODBC driver, you are not required but are encouraged to create a data source name first. You can do this through the ODBC Administrator. If you do create a data source name, you have to set the SAS environment variable MSSQLDSN to the created 'data_source_name'. Quotes are required if blank(s) or special characters are included.

To ensure that variable names generated by a Version 7 PROC SQL Pass-through view remain the same as they were in Version 6.x, provide the SAS option 'OPTIONS VALIDVARNAME=V6' before processing the existing V6 PROC SQL Pass-through view. If you

ignore this option, you may end up with variable names longer than 8 characters; SAS programs that are dependent on those names will no longer run as expected.

ODBC DRIVERS

In Version 6.x of the SAS System an Open Database Connectivity (ODBC) driver for the SAS System was shipped as part of base SAS. This driver provides data source updating and table creation via a local SAS session or a remote SAS/SHARE[®] server (in order to work with a SAS/SHARE server, SAS/CONNECT[®] or SAS/SHARE*NET[™] software must be installed). This driver has been updated for Version 7 of the SAS System. It now supports both Version 6 and Version 7 SAS/SHARE servers. When working with a Version 7 SAS/SHARE server, both long variable names and long member names are supported.

Since May 1997, a new product, the SAS Universal ODBC driver, has been available to SAS users. Unlike the SAS ODBC driver, a separate “shrinkwrap” product must be purchased for the Universal ODBC driver and only read-only access is supported. Unlike the ODBC driver that is shipped with base, the SAS Universal ODBC driver does not support Version 7 SAS/SHARE servers. However, the Universal ODBC driver does provide the following features that the base SAS ODBC driver does not:

- Support for MDB format files
- Support for FDB (CFO Vision Database) files
- Support for data sets that are stored on ASCII platforms
- Access to local data
- Access to remote data via FTP and HTTP

In addition, SAS does not have to be installed on the local PC in order for the SAS Universal ODBC driver to be usable.

CONCLUSION

Transferring data between the SAS System and other external data sources is and will continue to be a common activity in the business environment. Version 7 of the SAS System is designed to make the transfer process easier. With the addition of MS Access and Excel 97 to the Import/Export Facility, SAS users are given an even broader range of support for transferring data between SAS and other PC file formats and DBMSs. For further details about the options mentioned here and about additional features in SAS/ACCESS software that were not mentioned, refer to the SAS/ACCESS documentation.

ACKNOWLEDGEMENTS

The author wishes to thank Forrest Boozer, Michael Ho, Jana Van Wyk, Vino Gona, Tony Fisher, and John West for their invaluable assistance.

SAS, SAS/ACCESS, SAS/CONNECT, SAS/SHARE and SAS/SHARE*NET software are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. AS/400, DB2, IBM, and OS/2 are registered trademarks or trademarks of International Business Machines Corporation. ORACLE is a registered trademark of or trademark of Oracle Corporation.[®] Indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

Roger E. Sanders, SAS Institute Inc., 100 SAS Campus Dr.,
Cary, NC 27513, 919-677-8000, ext. 6330,
e-mail sasrys@unx.sas.com