Introduction to SAS/ACCESS® Software

Written and Prepared by

destiny
CORPORATION

Introduction

SAS/ACCESS software is a powerful tool in today's relational processing environments. Many companies around the world use SAS software as their tool of choice to extract information from an existing relational database. SAS Institute has created many versions of the SAS/ACCESS product to fluently communicate with popular database management systems found in these companies. This presentation discusses the use of SAS/ACCESS as a fundamental tool with today's relational databases. It is intended for users new to SAS/ACCESS software.

Multiple Engine Architecture In SAS

Using relational databases and external file formats in a SAS program is simple. It is just like referring to a SAS data set, however, the data comes from the relational database, instead of from within SAS. This is called Multiple Engine Architecture. It can provide for direct access to data without the need for copying into SAS, transparent access to the data, and database access without major physical storage requirements.

Field names, formats, and types are transparent between different environments

The two types of engines available with SAS are:

"Native" engines that read and write SAS data views, SAS data sets, and SAS transport files. The current native engines are:

SPSS

OSIRIS

BMDP

DEC Common Data Dictionary **(CDD)**

DEC Digital Table Interchange Format **(DTIF)**

"Interface" engines communicate with external environments such as relational databases. The current and future engines are:

DB2	(MVS)
CA-DATACOM/DB	(MVS)
Rdb/VMS	(VMS)
INGRES	(VMS,SunOS)
IMS-DL/I	(MVS)
ADABAS	(MVS)
SYSTEM2000	(MVS, CMS)
SQL/DS	(CMS, VSE)
ORACLE	(VMS, PRIMOS, AOS/VS, AIX, SunOS, HP-UX)
Database Manager	(OS/2)
dBASE .DBF file	(OS/2, Windows)
Lotus 123 .DIF file	(OS/2, Windows)
Prime Information	(PRIMOS)
AS/400	(OS/2, Windows)
VSAM	(MVS)
SYBASE	(SunOS, VMS)
CA/IDMS	(MVS)
Others	

Interface, ACCESS, DBLOAD

The SAS/ACCESS software consists of:

Interface Engine — This is the part that transfers data to the relational database and the SAS system. It provides transparent read and write abilities.

ACCESS Procedure — A Utility for defining ACCESS and VIEW descriptors that read and write information with relational databases.

DBLOAD Procedure — Used to create new tables, insert data, and to pass administrative SQL information to the DBMS.

Data Reading, Updating and Appending

Data reading, updating and appending can be done with the procedures below.

```
proc fsview data=view.one
```

```
proc fsedit data=view.one
```

```
proc append base=view.one
            data=view.two;
```

```
proc sql;
   update view.one set ...;
   delete from view.one where ...;
   insert into view.one values ...;
data view.one;
   modify view.one;
```

ACCESS and VIEW Descriptors

DBMS tables are given alias names that are used like SAS Data sets. For example,

```
proc sort data=view.one out=two;
```

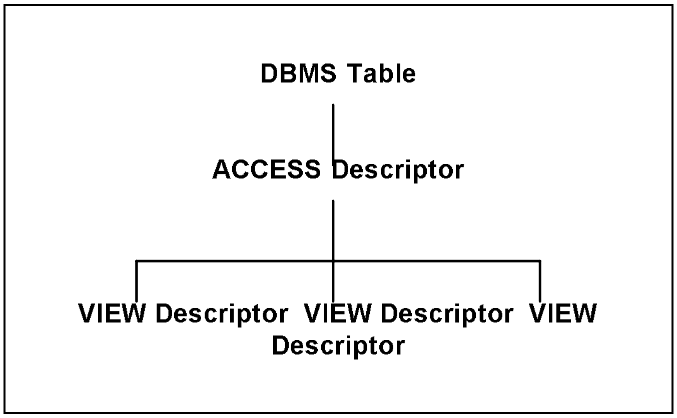
VIEW.ONE is the name of the view descriptor, which is the alias name that relates to the relational database. This view descriptor specifies what data to retrieve from the DBMS table and how to retrieve it. To use a view, simply precede the name of the information with a high level qualifier of what the library name or LIBNAME is. In the above example, it is VIEW.

Note: A SAS data set of the same name cannot exist in the same library.

A view descriptor relates to the specification of a DBMS table called an access descriptor. PROC ACCESS is used to create view and access descriptors. PROC DBLOAD creates only access descriptors.

ACCESS Descriptors

ACCESS descriptors are SAS files with a member type of ACCESS. They specify the columns available to SAS in a DBMS table and how information is accessed. ACCESS descriptors contain conversion information for relating DBMS names to SAS variables, types, and formats. They are usually used to create VIEW descriptors.



The DBMS is queried by the SAS system when an ACCESS descriptor is created and a VIEW descriptor is used, but not when a VIEW descriptor is created.

VIEW Descriptors

VIEW descriptors are SAS files with a member type of VIEW. They point to information stored in ACCESS descriptors and can specify a subset of the columns specified in an ACCESS descriptor. They can define different SAS variable names and formats than what is prescribed in the ACCESS descriptor. View descriptors are referenced as if they were a SAS data set and can specify the order which the information will be retrieved via indexing. When VIEW descriptors are referenced in a SAS program, information comes directly from the DBMS table. No intermediate files are created.

SQL Pass-Through Facility

SQL®, created by Dr. Codd of IBM is modeled by PROC SQL. PROC SQL, found in Base SAS, provides an alternate way to pass statements to the DBMS. You can connect to a DBMS(s) and disconnect from it, pass an SQL SELECT query that can be stored in a SAS data view, and can execute SQL statements available to the DBMS.

Note: Some SAS/ACCESS products have limited support of the PROC SQL Pass-Through Facility. Consult the SAS/ACCESS guide for your particular DBMS.

When to use the SQL Pass-Through Facility

- Joins with large tables from the same DBMS.
- SAS/AF applications doing transaction processing against a DBMS.
- When subsetting columns of a VIEW descriptor.
- When performing summaries of DBMS table information (COUNT, AVG, etc...)

The relation of SAS to SQL to standard Data Processing is simple:

SAS	SQL	Data Processing
SAS Data Set	Table	File
Observation	Row	Record
Variable	Column	Field

ACCESS Procedure

PROC ACCESS is available for creating ACCESS and VIEW descriptors, editing and browsing existing descriptors, and reading information from DBMS tables into SAS. ACCESS and VIEW descriptors can be created on-line, with program statements in batch, or interactive mode.