

CHAPTER

1

Overview of the SAS/ACCESS Interface to IMS

<i>Introduction to the SAS/ACCESS Interface to IMS</i>	3
<i>Purpose of the SAS/ACCESS Interface to IMS</i>	3
<i>Using the SAS/ACCESS Interface to IMS</i>	4
<i>How the IMS Engine and DATA Step Interfaces Differ</i>	4
<i>When to Use the IMS Engine Interface</i>	5
<i>When to Use the IMS DATA Step Interface</i>	5
<i>Features Not Supported by the IMS Engine Interface</i>	5
<i>Features Not Supported by the IMS DATA Step Interface</i>	5
<i>SAS/ACCESS Descriptor Files for IMS</i>	5
<i>Access Descriptor Files</i>	6
<i>View Descriptor Files</i>	6
<i>Executing SAS/ACCESS Programs in Batch Mode</i>	7
<i>Executing a Cataloged Procedure</i>	7
<i>DD Statements</i>	7
<i>Executing SAS/ACCESS Programs under TSO</i>	8
<i>Allocating Database Data Sets</i>	8
<i>About the Example Data in the Document</i>	8
<i>Running the Examples in This Document</i>	9

Introduction to the SAS/ACCESS Interface to IMS

This section introduces you to SAS/ACCESS software and briefly describes how to use the interface. This section also introduces the sample IMS data and SAS data files used in this document.

Purpose of the SAS/ACCESS Interface to IMS

SAS/ACCESS software provides an interface between SAS and the IMS database management system (DBMS). You can perform the following tasks with this SAS/ACCESS interface:

- create SAS/ACCESS descriptor files using the ACCESS procedure
- directly access data in IMS databases from a SAS program using the view descriptor files created with the ACCESS procedure
- extract data from IMS databases and place it in a SAS data file using the ACCESS procedure, the DATA step, or other SAS procedures
- update, insert, or delete data in IMS databases using the SQL procedure, SAS/FSP software, the APPEND procedure, or the MODIFY statement. The MODIFY

statement can be used with the IMS interface view engine, and supports REPLACE, DELETE, and INSERT calls.

- issue DL/I calls to update, insert, and delete data in IMS databases using the DATA step interface's INFILE, INPUT, FILE, and PUT statements.

Using the SAS/ACCESS Interface to IMS

The SAS/ACCESS interface to IMS consists of three parts:

- the ACCESS procedure, which you use to define the SAS/ACCESS descriptor files
- the IMS interface view engine, which enables you to use IMS descriptor files in SAS programs in much the same way you use SAS data files
- the DATA step interface, which enables you to access information in IMS databases using SAS programming statements.

The ACCESS procedure enables you to describe an IMS database to SAS in an access descriptor file. You can then create view descriptor files from the access descriptor file, which you can use in SAS programs in much the same way as you would use SAS data files. You can print, plot, and chart the data described by the view descriptor files, use it to create other SAS data sets, and so on. Chapter 3, “Defining SAS/ACCESS Descriptor Files,” on page 41 describes how to create and edit SAS/ACCESS descriptor files. Chapter 4, “IMS Data in SAS Programs,” on page 49 presents examples of using IMS data in SAS programs, and Chapter 5, “Browsing and Updating IMS Data,” on page 71 shows how to use the view descriptor files to update IMS data from within a SAS program.

The interface view engine is an integral part of the SAS/ACCESS interface, but the interface's design is transparent, so you are seldom aware of the engine. SAS interacts automatically with the engine when you use SAS/ACCESS view descriptors in your SAS programs, so you can use IMS data in your programs in much the same way as you use SAS data.

The DATA step interface provides special extensions of standard SAS INFILE and FILE statements to access IMS resources. Chapter 9, “How to Use the IMS DATA Step Interface,” on page 197 describes these statement extensions in detail.

You might need to combine data from several sources, including IMS databases, Version 6 SAS data sets, Version 7 SAS data sets, and other databases. With the SAS/ACCESS interface, such combinations are not only possible, but easy to do. SAS can differentiate among SAS data sets, SAS/ACCESS view descriptor files, and other types of SAS files, and it can use the appropriate access method.

How the IMS Engine and DATA Step Interfaces Differ

When comparing the two interfaces, you can identify some obvious differences:

- The IMS interface view engine requires you to create descriptor files. The engine uses information from the descriptor files to successfully attach IMS and retrieve or update the data being requested by the application. The DATA step interface requires no such files since it is a programming interface.
- The engine access method provides transparent access to IMS data. To access data, you simply make reference to a descriptor file and you have access to the data defined by the view. Coding DATA step programs requires in-depth knowledge of the database that is being accessed, and the ability to code host level calls to retrieve or update IMS data. You can, however, create SAS DATA step

views from the DATA step programs to provide users who are unfamiliar with the DL/I language transparent access to the data.

When to Use the IMS Engine Interface

Use the IMS engine interface for the following situations:

- transparent access to IMS data.
- access to data that lies in a single database path. Performance is enhanced when segment search arguments (SSAs) can be generated from WHERE statements.
- assignment of READ, WRITE and ALTER levels of protection with passwords.

When to Use the IMS DATA Step Interface

Use the IMS DATA step interface for the following situations:

- programs that need full control over DBMS access. The DATA step interface provides total CHKP control in an update program as well as control over DBMS calls and SSAs.
- transaction-style programs that need the capability of dynamically generating SSAs from SAS variable values in transaction files.
- multi-path processing or accessing data from multiple databases in the same application. Joining data can be more efficient when performed in the DATA step as opposed to the engine interface.
- access to message queues in a BMP region.
- access to Fast Path databases.

Features Not Supported by the IMS Engine Interface

The engine does not support Fast Path, message queue access, or HSSR. The SLI region type is also no longer supported; however, you can use the DBCTL feature of IMS/ESA and CICS/ESA for those functions.

Features Not Supported by the IMS DATA Step Interface

The DATA step does not support the DLITEST procedure and HSSR. The SLI region type is also no longer supported; however, you can use the DBCTL feature of IMS/ESA and CICS/ESA for those functions.

SAS/ACCESS Descriptor Files for IMS

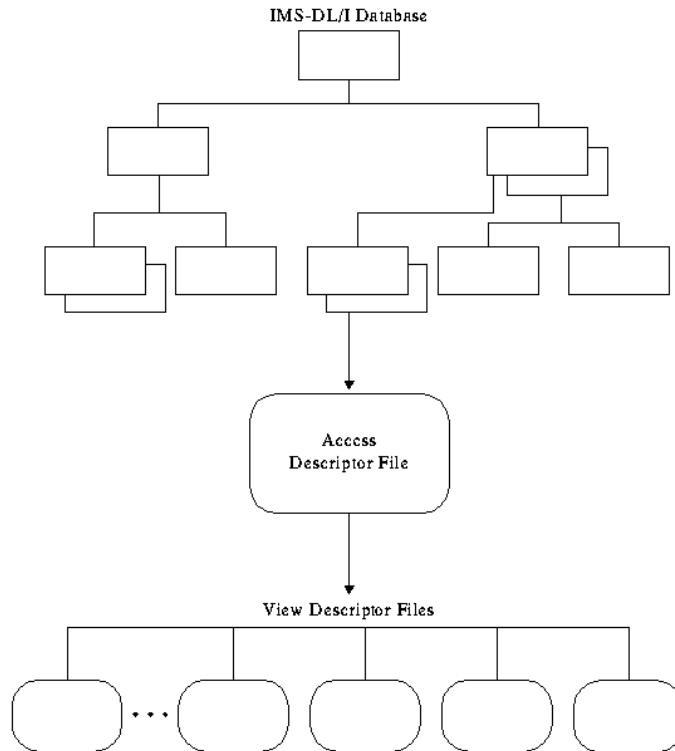
SAS/ACCESS descriptor files are the tools that the SAS/ACCESS interface view engine uses to establish a connection between SAS and IMS. To create these files, you run the ACCESS procedure using one of three methods:

- batch mode
- interactive line mode
- noninteractive mode.

There are two types of descriptor files: access descriptors and view descriptors. They are discussed in the next two sections. The following figure illustrates the relationships among an IMS database, an access descriptor, and view descriptors. Chapter 3,

“Defining SAS/ACCESS Descriptor Files,” on page 41 shows you how to create, browse, and edit these files.

Figure 1.1 Relationships between an IMS Database, an Access Descriptor, and View Descriptors



Access Descriptor Files

Access descriptor files are of member type ACCESS. Each access descriptor holds essential information about the IMS database that you want to access, for example, the name of the database; the names of segments, search fields, and key fields; segment lengths; names of non-search fields in the database; and data types. An access descriptor might also contain the corresponding SAS information, such as the SAS variable names and formats. Typically, you have only one access descriptor for each IMS database.

Since an access descriptor describes only one IMS database, you cannot create a single access descriptor that accesses multiple *physical* IMS databases; however, you can create a single access descriptor for a *logical* IMS database that accesses multiple physical databases.

View Descriptor Files

View descriptor files are sometimes called *views* because their member type is VIEW. This document uses the term *view descriptors* to distinguish view descriptor files from views created by the SAS SQL procedure or a DATA step.

Each view descriptor can define all the data in one path of the database, or a particular subset of the data in one path of the database (see Chapter 2, “IMS Essentials,” on page 11 for a discussion of paths and other IMS concepts). View

descriptors enable you to define all the data in one path of the database by selecting the items that you want to use and specifying selection criteria for only the records that you want. For example, you might want only records with a transaction date of July 3, 1995, and for customers who live in Richmond. You might have several view descriptors, each selecting different paths of data in an access descriptor that you have defined. You might also have view descriptors that select different subsets of data in one path of an access descriptor.

You can join data by using SAS's SQL procedure. With the SQL procedure, you can create a view that joins and summarizes data from multiple view descriptors (based on IMS databases), SAS data files, DATA step views, or other PROC SQL views. See Chapter 4, "IMS Data in SAS Programs," on page 49 for a discussion and examples that use the SQL procedure.

Executing SAS/ACCESS Programs in Batch Mode

The JCL (job control language) that is required to execute programs using the SAS/ACCESS interface to IMS in OS/390 batch mode is similar to that of other SAS jobs in OS/390 batch mode. Refer to the *SAS Companion for z/OS* for general information about SAS jobs in OS/390 batch environments.

Executing a Cataloged Procedure

The JCL for a batch job that accesses IMS data requires that you specify your site's designated cataloged procedure in the EXEC statement. So, instead of specifying your site's default SAS cataloged procedure (such as `// EXEC SAS`), you use the following EXEC statement:

```
// EXEC your-cataloged-procedure
```

The name of the cataloged procedure that invokes SAS and supports the use of the SAS/ACCESS interface to IMS differs at each installation, particularly if your installation uses separate cataloged procedures for accessing test databases and production DL/I databases. Be sure to check with your site's SAS support consultant for the correct procedure name. (SAS no longer supplies the Version 5 SASDLI cataloged procedure.) The installation notes that are shipped with the SAS/ACCESS interface to IMS explain to your database administrator how to create a cataloged procedure for your site.

The cataloged procedure for accessing IMS data contains all of the JCL statements and parameters that the SAS cataloged procedure contains, plus JCL statements and parameters necessary for the run-time execution of the IMS engine interface and IMS DATA step programs. PROC ACCESS can use the standard SAS catalog if the only task you are performing is creating descriptor files.

DD Statements

If you execute DL/I calls through a batch DL/I region (DLI or DBB), DD statements for all the database and index data sets that are accessed must be included in the job step JCL. DDnames and DS names (names of database data sets) must be obtained from the DBA staff at your site. Data sets that support the index must be allocated for the HIDAM database type.

If you execute DL/I calls through an online DL/I access region (BMP), database data sets are allocated to the DL/I control region or to the CICS control region. Therefore, you do not need to include DDnames for them in the job step JCL.

When you execute a batch DL/I region and want to log updates, you need a DD statement for a log data set. For information about using the IMSLOG option, on pointing to the IMS resident libraries, and information about pointing to the DBD, PSB, and ACBLIB, check with your DBA or refer to the installation instructions for the SAS/ACCESS interface to IMS.

All other JCL considerations that are outlined in the *SAS Companion for z/OS* apply to the IMS engine interface and IMS DATA step execution.

Executing SAS/ACCESS Programs under TSO

The SAS/ACCESS Interface to IMS can run interactively if your site has installed SAS under TSO. The TSO commands needed for the run-time execution of the IMS engine interface and IMS DATA step programs are similar to those for other TSO SAS jobs. (For general information about using SAS under TSO, see the *SAS Companion for z/OS* and the installation instructions for the SAS/ACCESS interface to IMS). PROC ACCESS can use the standard SAS CLIST if the only task that you are performing is creating descriptor files.

Allocating Database Data Sets

If you access DL/I databases through a batch DL/I region (DLI or DBB), you must first allocate the database data sets. You can allocate these database data sets from within or outside of SAS.

From within a SAS session, you use the SAS FILENAME statement to associate DDnames with database data sets and other OS/390 files that might be accessed by the interface view engine or a DATA step program. You can specify the FILENAME statements in the SAS Program Editor or in an AUTOEXEC file.

From outside of a SAS session, you can use a TSO ALLOCATE command. You can obtain the appropriate filerefs and data set names from the database administrator (DBA) staff at your site.

If you execute DL/I calls through an online DL/I access region (BMP), the database data sets are allocated to the respective DL/I control regions. Therefore, you do not need to allocate them with the TSO ALLOCATE command (or other means).

All other commands and TSO environment considerations are described in the *SAS Companion for z/OS*.

About the Example Data in the Document

This document uses two HDAM IMS databases, the ACCTDBD database and the WIRETRN database. These databases were created for a bank, and they contain data about the bank's customers and their checking and savings account transactions. The seven ACCTDBD database segments are named CUSTOMER, CHCKACCT, CHCKDEBT, CHCKCRDT, SAVEACCT, SAVEDEBT, and SAVECRDT. The WIRETRN database has one segment, WIRETRAN, and includes only data pertaining to wire transfers of money. All the data in the document is fictitious.

The document also uses one HIDAM database, EMPLINF2, in the examples.

Note: These databases are designed to show how the interface treats IMS-DL/I data. They are not meant as an example for you to follow in designing databases for any purpose. △

Appendix 2 gives more information about the ACCTDBD database and the data it contains. Appendix 2 also includes definitions of all the view descriptors referenced in this document and all the SAS data files and statements used to create them. Chapter 3, “Defining SAS/ACCESS Descriptor Files,” on page 41 provides information about the WIRETRN database and definitions of the view descriptors used in the examples.

Running the Examples in This Document

To run the examples based on the ACCTDBD and WIRETRN databases, you must first load the database files and define the access and view descriptors shown in Appendix 2. Use the sample library files described here.

IMSLD

contains the source programs for loading the ACCTDBD, EMPLINF2, and WIRETRN database files for both the engine interface and DATA step. It includes the JCL used to allocate the IMS databases, to create DBDs and PSBs, and to create needed flat files.

IMSEX

contains the example SAS programs that use the engine interface, as shown in Chapter 4, “IMS Data in SAS Programs,” on page 49 and Chapter 5, “Browsing and Updating IMS Data,” on page 71.

IMSDS

contains the example SAS programs that use the DATA step interface, as shown in Chapter 1, “Overview of the SAS/ACCESS Interface to IMS,” on page 3 and Chapter 9, “How to Use the IMS DATA Step Interface,” on page 197.

