APPENDIX

1

Information for the Database Administrator

Introduction **119**
How the SAS/ACCESS Interface to SYSTEM 2000 Software Works **119**
How the SYSTEM 2000 Interface View Engine Works **120**
Calls Made on Behalf of the ACCESS Procedure **121**
Calls Made on Behalf of the DBLOAD Procedure **121**
Calls Made on Behalf of the QUEST Procedure **121**
Calls Made by Other SAS Procedures **121**
Effects of Changing a SYSTEM 2000 Database Password **122**
How Changing a Database Definition Affects Descriptor Files **122**
Changes That Have No Effect on Existing View Descriptors **122**
Changes That Might Have an Effect on Existing View Descriptors **123**
Changes That Cause Existing View Descriptors To Fail **123**
Ensuring Data Security **123**
SYSTEM 2000 Security **123**
SAS System Security **124**
Enabling the Rollback Log **124**
Levels of Locking **125**
Maximizing SYSTEM 2000 Performance **125**

Introduction

This appendix explains how the SAS/ACCESS interface to SYSTEM 2000 software works so that you can decide how to administer its use at your site. It also includes topics about data security, database recovery, levels of locking, and maximizing SYSTEM 2000 performance.

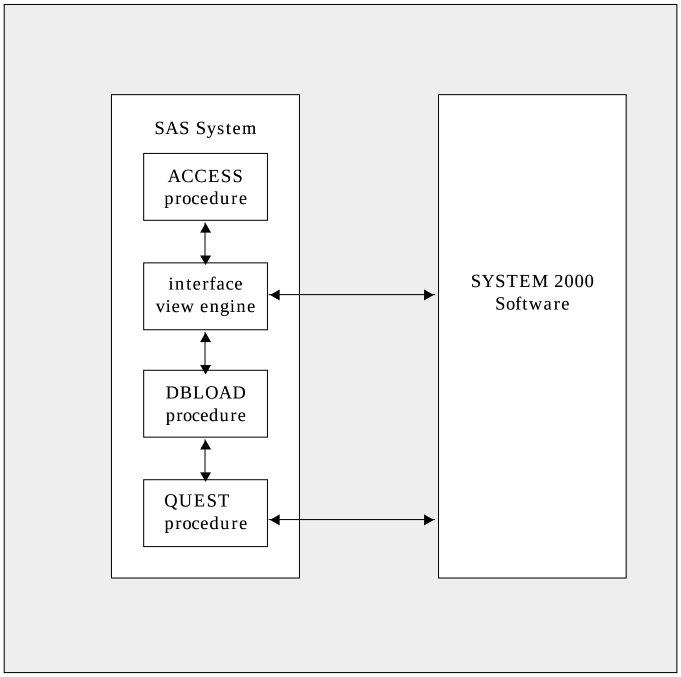
How the SAS/ACCESS Interface to SYSTEM 2000 Software Works

When you use the ACCESS procedure to create an access descriptor file, the SAS System calls SYSTEM 2000 software to get a description of the database. When you create a view descriptor file, the SAS System has information about the database in the access descriptor, so it does not call SYSTEM 2000 software.

The ACCESS procedure writes the descriptor files to a SAS data library. Then, when you use a SAS procedure with a view descriptor whose data is in a SYSTEM 2000 database, the SAS System Supervisor calls the interface view engine to access the data. The engine can access a database for reading, updating, inserting, and deleting.

The connections between the SAS/ACCESS procedures and the SYSTEM 2000 software are shown in Figure A1.1 on page 120.

Figure A1.1 How the SAS System Connects to SYSTEM 2000 Software



How the SYSTEM 2000 Interface View Engine Works

The interface view engine is a PLEX (Programming Language Extension) applications program, which retrieves and updates data in a SYSTEM 2000 database. Calls to the engine are in one of the following categories:

- □ calls made on behalf of the ACCESS procedure when it is creating an access descriptor
- □ calls made on behalf of the DBLOAD procedure that reference a view descriptor with the VIEWDESC= option
- □ calls made on behalf of the QUEST procedure to negotiate an execution environment
- □ calls made by a SAS DATA step or by SAS procedures that reference a view descriptor with the DATA= option.

In all situations, the same PLEX commands initiate and terminate communication between the interface view engine and SYSTEM 2000 software. Each time a different SAS procedure requires use of SYSTEM 2000 software, it makes an initialization call to the engine. This first call establishes communication with SYSTEM 2000 software and opens the requested database.

The engine determines whether it must issue the START S2K command for a single-user job or the Multi-User environment, as specified by the calling SAS procedure. Then it issues an OPEN command for the specified database and returns control to the procedure. Additional calls to the engine perform retrieval and update operations required by the SAS procedure.

When the SAS procedure terminates, the engine issues a CLOSE command for the database that was opened. It issues the STOP S2K command only when the entire SAS session terminates, or when you want to run the QUEST procedure in the same environment (single user or Multi-User) that SAS programs have been running in.

Calls Made on Behalf of the ACCESS Procedure

The ACCESS procedure calls the interface view engine to retrieve item and record information for a specified database. The engine sends the component number, name, type, picture, level number, and key status (along with the database cycle number and so on) back to the procedure for each item and record in the database. The procedure stores this information in the access descriptor for later use when creating view descriptors.

Also, if you are using the ACCESS procedure to extract information into a SAS data file, the ACCESS procedure calls the engine.

Calls Made on Behalf of the DBLOAD Procedure

When you create a new database, the DBLOAD procedure always creates an access descriptor and a view descriptor.

When you insert data into an existing SYSTEM 2000 database, you must specify an appropriate view descriptor in the DBLOAD procedure. The view descriptor provides a mapping between the SAS variables containing data to be inserted and the SYSTEM 2000 components that will be used to insert the data into the database. It also contains the database name, password, and access mode to be used when you do inserts.

For each observation that is retrieved from the data file specified on the DATA= option, a corresponding call is made to the interface view engine. The engine inserts the data into the database identified by the view descriptor. The engine uses only insert mode (one at a time) for inserting new descendant records into existing logical entries. Also, if the DATA= option identifies a SYSTEM 2000 view descriptor, the interface view engine is called to read that view.

When you load new logical entries (starting with records at level 0) into a database, you can specify that you want to use an optimized loading process. SYSTEM 2000 software processes the new logical entries as one batch of inserts (PLEX load mode). This optimized load mode is faster than inserting records one at a time. However, it causes the software to attach the database with exclusive use and no other database can be open in the same execution environment. Therefore, if your job is using optimized load, your input cannot also be a SYSTEM 2000 view descriptor of a database in the same environment.

Calls Made on Behalf of the QUEST Procedure

Usually the QUEST procedure communicates directly with SYSTEM 2000 software. When you enter SYSTEM 2000 statements (commands), they are processed immediately, and the results are sent back to you interactively. However, there is one exception. Each time you execute the QUEST procedure, it calls the engine to request permission to execute. If no other SAS programs are using the engine in the same environment, permission is granted. Otherwise, permission to execute is refused. Similarly, when the QUEST procedure terminates, it calls the engine to signal the event so that the engine will allow other SAS programs to execute. The engine does not allow SAS jobs to execute in the same environment while the QUEST procedure is running.

Calls Made by Other SAS Procedures

SAS procedures can access records in a SYSTEM 2000 database by referring to a view descriptor with the DATA= option. The SAS System examines the view descriptor to determine which database management system is referred to and passes control to the appropriate engine. The interface view engine uses information stored in the view descriptor (for example, access mode, password, database name, component numbers, levels, types, and so on) to process SYSTEM 2000 data records as if they were observations in a SAS data file.

Before doing any retrievals, the engine processes a PLEX dynamic where-clause (if specified) to select a subset of data records that are to be processed as observations. The engine constructs the dynamic where-clause from the view where-clause and the SAS WHERE clause (if any). If no view where-clause exists, a where-clause is constructed to locate all database records.

The dynamic where-clause processing returns a Locate File containing the addresses of database records that satisfied the selection criteria. Based on those addresses, the engine issues a combination of GET S2KCOUNT, GET1, and GETA commands to read one or more database records. Then the engine combines data from the records (according to the view descriptor) to form a SAS observation, which it passes back to the calling procedure for processing.

Based on the capabilities of the SAS procedure, the next call to the engine might be a request to update or delete the SAS observation that was just retrieved. For updates, the engine issues MODIFY, INSERT, REMOVE and so on for one or more data records, depending on how many records were used to construct the observation. Typically, the SAS procedure then calls the engine again to retrieve another SAS observation. If so, the engine locates another group of records, constructs another SAS observation, and returns it to the SAS procedure. This cycle continues until the SAS procedure terminates or until the last qualified SAS observation has been constructed and returned to the SAS procedure. In addition, the engine uses other commands, such as COMMIT, ROLLBACK, and CLEAR, to control processing.

Effects of Changing a SYSTEM 2000 Database Password

The master password holder can change any database password at any time. If you change a password used in a view descriptor, you can either change the view descriptor or override the stored password each time you need to use the view descriptor. The software does not care whether you use the same password. The engine only needs to have you use a password that has enough authority to service the view descriptor.

Since SYSTEM 2000 passwords are not stored in access descriptors, there are no effects if a password is changed. Passwords can be stored in view descriptors, but changing a password does not affect a view descriptor. It still has all items, but you might not be able to use the view. Nothing is automatically changed in the descriptors when you change a password or its authorities.

How Changing a Database Definition Affects Descriptor Files

As mentioned in Chapter 7, "ACCESS Procedure Reference," on page 73, changes to a database definition can affect descriptor files. Before changing a definition, consider the following guidelines.

Changes That Have No Effect on Existing View Descriptors

The following changes to a SYSTEM 2000 database definition have no effect on existing view descriptors:

- □ creating or deleting indexes.
- □ inserting new schema items.
- □ deleting schema items not referenced in any view descriptor. (Note that if an access descriptor includes the deleted item, users could eventually create a view descriptor using that item, which would be a problem.)

- inserting or deleting schema records in paths not referenced in any view descriptor. (Note that if an access descriptor references the changed path, users could eventually create a view descriptor that would have a problem.)

Changes That Might Have an Effect on Existing View Descriptors

The following changes to a SYSTEM 2000 database definition might have an effect on existing view descriptors:

- changing an item name. If the item name was used in a where-clause or an ordering-clause stored in the view descriptor, you will receive a syntax error message, indicating an unrecognized component name, when you try to use the view descriptor.
- changing the attributes of items not in the view descriptor but referenced in the stored where-clause.

Changes That Cause Existing View Descriptors To Fail

The following changes to a SYSTEM 2000 database definition cause existing view descriptors to fail when they are used:

- inserting or deleting a level in the path of a view descriptor.
- changing the attributes of an item or the component number of an item, so that it points to something different. Specifically,
  - You can change the pictures for CHARACTER, TEXT, and UNDEFINED item types, but you cannot change them to a date or numeric item type.
  - You cannot change a DATE item type to any other item type.
  - You cannot change a numeric item type to a non-numeric item type, and you cannot change its picture.
- changing the component number of parent records for any schema item or record in the path of a view descriptor.
- deleting items that are referenced in a view descriptor.

The interface view engine validates the view against the current definition and issues an informative error message if it detects discrepancies.

Ensuring Data Security

The SAS System preserves the data security provided by SYSTEM 2000 software and the SAS System. As the DBA, you have control over who has SYSTEM 2000 authorities. You control who can create SYSTEM 2000 databases, and creators of the databases control who can access the data. Therefore, SAS System users can access only SYSTEM 2000 databases they created or ones for which they have been granted specific password authorities.

To secure data from accidental update or deletion, you can take precautionary measures on both sides of the interface.

SYSTEM 2000 Security

On the SYSTEM 2000 side of the interface, give users only the authority they must have with secondary passwords. For example, JANE needs to be able to create a view descriptor that reads and selects just the personal information about each employee in the EMPLOYEE database, which is stored in the ENTRY record. JANE can only do retrievals and where-clause selection on schema items C1 through C16. Use the following commands to assign her authorities:

```
valid password is jane;
assign r,w to c1 through c16 for jane;
```

JOHN needs to add new employees to the data base, so he needs all authorities:

```
valid password is john;
assign r, u, w to all components for john;
```

With retrieval, update, and where-clause authorities, JOHN can create a view descriptor that reads the data records for the schema items and can use that view descriptor to add new logical entries to the database.

If the SYSTEM 2000 software detects a security violation while a SAS procedure or DATA step is running, it issues a return code 45 or 47 and an error message. Partial updates will be rolled out (cancelled), if rollback is enabled for the database.

SAS System Security

To secure data from accidental update or deletion, you can do the following on the SAS System side of the interface:

□ Set up all access descriptors yourself and drop items containing sensitive data.

□ Set up all view descriptors yourself and give them to users on a selective basis. You can store the appropriate passwords in the descriptors, or you might not want to store any passwords so that the user must supply his password.

□ Give users read-only access to the SAS data library where you store the access descriptors. Read-only access prevents users from editing access descriptors and allows them to see only the items selected for each view descriptor.

□ Set up several access descriptors for multiple secondary passwords, or require the user to create them.

Enabling the Rollback Log

A single SAS observation can be composed of one or more SYSTEM 2000 database records. Therefore, a single SYSTEM 2000 update command to update a given observation might involve several internal SYSTEM 2000 update commands. If one of these update commands fails after several others have already succeeded, the status of the entire update is incomplete.

In order to guarantee the data integrity, you must enable the rollback feature. You can do this easily with the QUEST procedure, by issuing the ENABLE ROLLBACK statement from the CONTROL language. When you enable rollback, you must be sure that the Rollback Log (database File 8) and the Update Log (database File 7) are allocated. With rollback enabled, the SYSTEM 2000 software can roll back the database to its status just prior to the sequence of commands that triggered the error.

If rollback is not enabled, partial updates can occur in the event of error return codes. Errors could arise from security violations or from bad data, for example, data that do not match the SAS informat or that have too many significant decimal places for a specific item's numerical precision. Also, if LHOLD=YES is specified in the SYSTEM 2000 execution parameters and rollback is not enabled, the interface view engine could receive return code 111, which causes an update to be rejected.

Levels of Locking

SAS software supports several levels of locking through the CNTLLEV data set option. If CNTLLEV equals REC (the default), SYSTEM 2000 software performs record level locking. The interface view engine interprets any CNTLLEV setting other than REC to signify that it should do an exclusive open of the database. Also, the database is under exclusive use if you have issued the S2KLOAD statement in the DBLOAD procedure. Exclusive use locks out all other users entirely until the database is closed, which normally occurs when the procedure ends. (The database close operation depends on the procedure.)

Note: Some SAS procedures set CNTLLEV equal to MEM internally for their own processing reasons. Many statistical procedures must make multiple passes of the data. For example, finding the median requires more than one pass. △

In a Multi-User environment, exclusive use of a database can cause undesirable contention for a database. Also, if you have specified optimized load mode (S2KLOAD) in the DBLOAD procedure, your input to that load cannot be a SYSTEM 2000 view descriptor for a database in the same environment.

When exclusive use of the database is not requested, the interface view engine uses SYSTEM 2000 record-level locking and multiple local holds. Basically, an observation is locked for retrieval and unlocked only when some other observation is retrieved or when the file is closed. Updates do not unlock the observation. This kind of locking could cause contention in a SYSTEM 2000 database. The interface view engine takes the following steps to keep the contention to a minimum:

- □ At retrieval time, the engine attempts to lock all records in the path (with the PLEX /HOLD option). If the lowest level record in the path (that is, the deepest one, farthest from level 0) cannot be locked, an error return code is sent back to the SAS System, indicating that this observation cannot be locked. Records above the bottom level in the view will be locked if possible, but the engine does not regard it as an error when they cannot be locked, and no message is given to the user.
- □ At update time, only those records that were successfully locked can be updated. For updates at levels that were not previously locked, the engine tries again to obtain the locks. If it cannot get them, the update fails, with a return code indicating that it could not get the necessary lock; partial updates are rolled back, assuming that rollback is enabled. If the engine gets the locks, it checks to see if the data in the database are still as they were when the data were originally retrieved; if they are the same, the update takes place.

The purpose of this locking mechanism is to avoid contention. You can always access a path as long as the bottom record can be locked. You will not have to wait for someone else to drop a lock on one of the upper level records. (However, there might be relatively few locks of upper level records.) You are guaranteed to be able to update only items in the record at the bottom of the view descriptor. The engine will tolerate attempts to update any level and will honor the update if it can.

Maximizing SYSTEM 2000 Performance

Among the factors that affect SYSTEM 2000 performance are the size of the database being accessed, the number of items being accessed, and the number of data records qualified by the selection criteria. For databases that have many items and many data records, you should evaluate all SAS programs that need to access the database directly. In your evaluation, consider the following questions:

□ Does the program need all the SYSTEM 2000 items? If not, create and use an appropriate view descriptor that includes only the items to be used.

□ Do the selection criteria retrieve only those data records needed for subsequent analysis? If not, specify different conditions so that the selected records are restricted for the program being used.

□ Are the data going to be used by more than one procedure in one SAS session? If so, consider extracting the data and placing them in a SAS data file for SAS procedures to use, instead of allowing the data to be accessed directly by each procedure. See “Performance Considerations” on page 40 for circumstances when extracting data is the more efficient method.

□ Do the records need to be in a particular order? If so, include a SYSTEM 2000 ordering-clause in the appropriate view descriptors or a SAS ORDER BY clause in a SAS program.

□ Do the selection criteria allow SYSTEM 2000 software to use key (indexed) items and non-key (not indexed) items efficiently? See “SYSTEM 2000 Where-Clause” on page 90 for some guidelines on specifying efficient selection criteria.

□ What kind of locking mechanism will SYSTEM 2000 software need to use? (See “Levels of Locking” on page 125.)

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/ACCESS® Interface to SYSTEM 2000® Data Management Software: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/ACCESS® Interface to SYSTEM 2000® Data Management Software: Reference, Version 8

ISBN 1–58025–549–3

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

The Institute is a private company devoted to the support and further development of its software and related services.