Criteria for Moving a SAS Application to SYSTEM 2000 Data Management Software
Carl F. Carpenter, SAS Institute Inc., Austin, TX

ABSTRACT
SYSTEM 2000 Data Management Software is currently available to SAS users on IBM® or IBM compatible mainframes only. Many SAS applications can take advantage of SYSTEM 2000 software options to improve processing performance, DASD utilization, and access to their SAS data. With the availability of the SAS/ACCESS® interface to SYSTEM 2000 software in Version 6 of the SAS System, there are virtually no limitations on access to data stored in a SYSTEM 2000 database. This paper specifies criteria that enable you to identify SAS applications as candidates for utilizing SYSTEM 2000 software as well as the steps that must be performed to accomplish the move of the applications to a data structure based on SYSTEM 2000 software.

INTRODUCTION
The purpose of this paper is to familiarize you with the enhancements available in the Version 6 SAS System that make the methods of accessing and retrieving data transparent to you. I will supply you with some general information on how storing data in a SYSTEM 2000 database can provide measurable performance improvements and increased data integrity. Using this information, I will provide you with some guidelines on how to determine which SAS applications are best suited for movement into a SYSTEM 2000 database structure. After discussing why it makes sense to move applications to SYSTEM 2000 databases and how to determine which applications are best suited for SYSTEM 2000 software, I will discuss various procedures which enable you to move data within a SAS data set to a SYSTEM 2000 database. Finally, I will discuss the specifics of accessing your data from virtually any procedure, including the DATA step, utilizing SYSTEM 2000 software.

WHY WOULD YOU WANT TO USE SYSTEM 2000 SOFTWARE?

The SAS System is well renowned for its ability to handle all forms of computerized data. The DATA step can read any data format that is available on computers and can create SAS data sets or end-user reports from this input. Users have come to rely on this power and flexibility to help them complete all forms of job tasks, such that the SAS System is no longer used exclusively as a statistical modeling tool.

The underlying method of retrieving SAS data is sequential. In Version 6 of the SAS System, the base I/O (Input/Output) engine now supports index items for selection to help reduce the amount of I/O required to access your data.

Sequential access methods are very efficient when a majority of the data in the data set is actually needed by the process. As the amount of data needed by a particular application decreases, the benefits afforded by access methods other than sequential become much more important, as is the ability for your software to utilize indexes.

SYSTEM 2000 software offers a variety of options that can further enhance the data processing methods used by the SAS System. Specifically, SYSTEM 2000 software can reduce DASD (disk space) requirements, enhance access to data, and in some cases, provide improved performance of I/O. Additionally, SYSTEM 2000 software provides extended functionality to the user. These features include extensive security options, data integrity features, an English-like query language, and a complete Multi-User® software facility for implementing real-time data sharing.

Improved I/O Performance
With Version 6 of the SAS System, all procedures now have a global option of WHERE, which enables you to avoid costly preprocessing of data in order to subset your data for a particular procedure. Although the specification of an indexed item in a WHERE option is not required, utilizing indexed items can reduce resource utilization required to access your data. When you utilize the SAS/ACCESS interface to SYSTEM 2000 software, you can gain some additional performance improvements over the base Version 6 I/O engine.

In Version 6, all I/O engines enable you to specify multiple Boolean conditions in the global WHERE option to subset your data prior to procedure execution time. The items contained within the expression may or may not be indexed. The base engine I/O analyzes the items in your WHERE conditions to determine whether to access the data via an index or to make a sequential pass of the data. The base I/O engine only uses one indexed item to subset data prior to procedure execution time, and that index may be connected to other conditions only with the Boolean operator AND, with the exception of selecting based on multiple values for a single item. The following is an example of a WHERE option using an indexed item called ITEM1 that is reduced to selection based on one indexed item:

```
WHERE item1='a' or item1='b' or item1='c'
```

In the next series of commands, if ITEM2 is also an indexed item, the first two commands result in the base engine using either ITEM1 or ITEM2 as an index. The third example results in a sequential pass of the data.

```
WHERE item1='a' or item1='b' or item1='c'
WHERE item1='a' and item2='a'
WHERE item1='a' or item1='b' and item2='b'
WHERE item1='a' or item2='a'
```

If the engine determines that index processing is more efficient than a sequential pass of the data, a logical subset of data is created using the index. This logical subset of data is then passed sequentially during procedure execution to resolve all other conditions that you specified in the WHERE option.

Although the introduction of indexes into the Version 6 base I/O engine greatly enhances access to your data, in some cases, SYSTEM 2000 software can provide additional performance improvements above those provided by the base I/O engine. When many items are used in a selection expression, the SYSTEM 2000 engine utilizes all of the items as indexes and utilizes indexed processing with both AND and OR Boolean connectors to subset data before procedure execution. This results in less data being passed to the procedure during execution, and therefore, in many cases, fewer computer resources are utilized. Note that the efficiencies available through the use of SYSTEM 2000 software usually improve performance only when multiple indexed items are specified in the WHERE option, and each index greatly narrows the set of records that eventually qualify.

Reduced DASD Requirements
Another area where SYSTEM 2000 software can enhance resource utilization is DASD (disk space) utilization. The Version 6 base engine has a data set option called COMPRESS. If you use this...
option, your data set will generally occupy a minimum of disk space.

In two cases, SYSTEM 2000 software can reduce this space further: by using indexes on variable length character items and by eliminating data redundancy.

Within the SAS System, character data are defined as fixed length fields of data within an observation, and every observation must carry the maximum field length for possible data values without regard to actual usage. When you specify COMPRESS=YES for a data set, each observation is viewed as a byte string, and all occurrences of 3 or more identical bytes are compressed.

Using SYSTEM 2000 software, you can define a field to be smaller than the actual required length. Significant characters beyond the defined length are stored in an overflow table. Efficient pointers to this data make the retrieval process for the overflowing values minimal.

One method that SYSTEM 2000 software uses to reduce space consumption is to store data that overflow only once for an indexed item, even though the value may be contained in many observations.

Figure 1 shows how storing overflow data only once can have a significant impact on DASD usage. Consider a data set containing 100,000 observations. VENDOR is an item within each observation and is defined to be 32 characters in length. Assume that you have 100 unique VENDOR values in your data set. The average length of the values is 16, and the SYSTEM 2000 defined length is 4. Figure 1 shows how SYSTEM 2000 storage techniques can have a significant impact on DASD usage.

| Base I/O Engine | Number of Obs | 100,000 |
| Storage          | Defined Size  | 32 characters |
|                  | Average Size  | 16 characters |
|                  | Compressed Size | 1.8M characters |

SYSTEM 2000

| Number of Obs | 100,000 |
| Storage       | Defined size | 4 characters |
|               | Average size  | 16 |
|               | Average occurrence | 1,000 per unique value |
|               | Data in overflow | 1,000 characters |
|               | Data in pointers | 400,000 characters |
|               | Total space     | 1.4M characters |

Net savings for the data set is 1.4M characters of space.

Figure 2 shows how storing overflow data only once can have a significant impact on DASD usage. Consider a data set containing 100,000 observations. VENDOR is an item within each observation and is defined to be 32 characters in length. Assume that you have 100 unique VENDOR values in your data set. The average length of the values is 16, and the SYSTEM 2000 defined length is 4. Figure 1 shows how SYSTEM 2000 storage techniques can have a significant impact on DASD usage.

Figure 3 shows how storing overflow data only once can have a significant impact on DASD usage. Consider a data set containing 100,000 observations. VENDOR is an item within each observation and is defined to be 32 characters in length. Assume that you have 100 unique VENDOR values in your data set. The average length of the values is 16, and the SYSTEM 2000 defined length is 4. Figure 1 shows how SYSTEM 2000 storage techniques can have a significant impact on DASD usage.

<table>
<thead>
<tr>
<th>OBS</th>
<th>OWNER</th>
<th>DEPT</th>
<th>NAME</th>
<th>SERIALNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B. JONES</td>
<td>XCS</td>
<td>ENETTOOL</td>
<td>BB00306H</td>
</tr>
<tr>
<td>2</td>
<td>B. JONES</td>
<td>XCS</td>
<td>RHJ PCRT</td>
<td>TE021666H</td>
</tr>
<tr>
<td>3</td>
<td>B. JONES</td>
<td>XCS</td>
<td>RHJ PCRT</td>
<td>23-006818</td>
</tr>
<tr>
<td>4</td>
<td>B. JONES</td>
<td>XCS</td>
<td>RHJ PCRT</td>
<td>23-006818</td>
</tr>
<tr>
<td>5</td>
<td>B. JONES</td>
<td>XCS</td>
<td>RHJ PCRT</td>
<td>26-0017012</td>
</tr>
<tr>
<td>6</td>
<td>B. JONES</td>
<td>XCS</td>
<td>RHJ PCRT</td>
<td>26-0017012</td>
</tr>
<tr>
<td>7</td>
<td>B. JONES</td>
<td>XCS</td>
<td>RHJ Y75</td>
<td>00B1000428</td>
</tr>
<tr>
<td>8</td>
<td>B. JONES</td>
<td>XCS</td>
<td>RHJ3270</td>
<td>531086</td>
</tr>
<tr>
<td>9</td>
<td>B. TALLEY</td>
<td>XTS</td>
<td>CCTPC</td>
<td>A00531003533</td>
</tr>
<tr>
<td>10</td>
<td>B. TALLEY</td>
<td>XTS</td>
<td>CTT PC</td>
<td>0094393</td>
</tr>
<tr>
<td>11</td>
<td>B. TALLEY</td>
<td>XTS</td>
<td>CTT PC</td>
<td>0094393</td>
</tr>
<tr>
<td>12</td>
<td>B. TALLEY</td>
<td>XTS</td>
<td>CTT PC</td>
<td>0415987</td>
</tr>
<tr>
<td>13</td>
<td>B. TALLEY</td>
<td>XTS</td>
<td>CTT PC</td>
<td>367263</td>
</tr>
<tr>
<td>14</td>
<td>B. TALLEY</td>
<td>XTS</td>
<td>CTT PC</td>
<td>367263</td>
</tr>
<tr>
<td>15</td>
<td>B. TALLEY</td>
<td>XTS</td>
<td>CTT STEP</td>
<td>46893</td>
</tr>
<tr>
<td>16</td>
<td>B. TALLEY</td>
<td>XTS</td>
<td>CTT3270</td>
<td>532239</td>
</tr>
</tbody>
</table>

Figure 2 Listing of Inventory Data Using SAS Software

* OWNER  DEPT  NAME  SERIALNO
* B. JONES  XCS  ENETTOOL  BB00306H
* B. JONES  XCS  RHJ PCRT  TE021666H
* 23-006818
* 23-006818
* 26-0017012
* 26-0017012
* 00B1000428
* B. JONES  XCS  RHJ3270  531086
* B. TALLEY  XTS  CTT PC  A00531003533
* 0094393
* 0094393
* 0415987
* 367263
* 367263
* 46893
* 532239

Figure 3 Listing of Inventory Data Using SYSTEM 2000 Software

Consider the following definitions for an INVENTORY system and a HELP system:

**INVENTORY Definition**

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Location information</td>
</tr>
<tr>
<td>1</td>
<td>Group information</td>
</tr>
<tr>
<td>2</td>
<td>Detail information</td>
</tr>
<tr>
<td>3</td>
<td>Documentation information</td>
</tr>
</tbody>
</table>

**HELP Definition**

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Problem occurrence</td>
</tr>
<tr>
<td>1</td>
<td>Problem bypass</td>
</tr>
<tr>
<td>2</td>
<td>Status</td>
</tr>
<tr>
<td>3</td>
<td>Hardware request</td>
</tr>
</tbody>
</table>

The definition for each system contains information that can be logically grouped together. The definition of each system has data arranged such that each group of data is at a level within the hierarchy which allows for the greatest reduction of data redundancy. In the HELP definition, the hardware request information was placed at the bottom of the hierarchy even though it is related to information in the first level. There will be only one hardware request record per level 0 record. Making the hardware request information the last level achieves the greatest effect on reducing space requirements because less than 5 percent of all entries in the HELP system resulted in a hardware request being made. SYSTEM 2000 software does not require space for the record lowest in the hierarchy unless the record contains values.
Figure 4 shows an INVENTORY system where each piece of equipment starts out with no documentation and grows to where each piece of equipment has nine documents attached. The following is true:

- Owners of equipment - Level 0 - 187 records
- Groups of equipment - Level 1 - 301 records
- Pieces of equipment - Level 2 - 1013 records

Figure 4 shows the increasing benefits of using SYSTEM 2000 when your data grow at the lowest level.

<table>
<thead>
<tr>
<th>INVENTORY</th>
<th>Number of Obs</th>
<th>Number of Records</th>
<th>Number of Bytes</th>
<th>Number of Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS</td>
<td>1329</td>
<td>74</td>
<td>184K</td>
<td>5</td>
</tr>
<tr>
<td>S2K</td>
<td>5102</td>
<td>4126</td>
<td>1505K</td>
<td>41</td>
</tr>
<tr>
<td>SAS</td>
<td>8178</td>
<td>8178</td>
<td>2826K</td>
<td>77</td>
</tr>
<tr>
<td>S2K</td>
<td>2508K</td>
<td>67</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4 Effects of Data Growth

Figure 5 shows statistics pertaining to the HELP system. In this system, each reported problem has approximately four activities logged against it. As you can see, where the INVENTORY system required about 1300 observations, the HELP system achieves better space utilization consistent at about 8%. As you can see, if your data occur naturally in a one-to-many relationship, you will see space savings that will grow proportionately with how large the many factor is.

Enhanced Access to Your Data

A significant access enhancement available to you is the ability to share data in either browse or update mode. There are many applications where it is very critical to have access to the latest information. SYSTEM 2000 software provides you with the ability to maintain your data in a real-time environment.

SYSTEM 2000 software provides various levels of data locking. Data locking is the ability of the software to control access to data by several concurrent users. With the data-locking features of SYSTEM 2000 software, you always have the latest copy of an observation, regardless of how many other users are attempting to access or update the observation.

The Multi-User feature of SYSTEM 2000 software is required to implement shared data access. Multi-User software is a control program that can be executed in either an MVS address space or a CMS virtual machine. You will access the software from different MVS address spaces or CMS virtual machines. Special interaddress space communication vehicles are utilized that enable the user region and the Multi-User control program to interchange requests and data.

The features of Multi-User software give you the following benefits:

- You can reduce costs associated with having to preprocess data in order to create copies for multiple users.
- Information is more timely and accurate.
- You are able to more closely control access to your data.

Increased Data Integrity

Data integrity becomes important when real-time concurrent updating of shared data is implemented with software such as Multi-User. One of the most critical issues regarding data integrity is the ability to recover from various types of outages and user errors. SYSTEM 2000 software enables you to use two methods of recovery, either separately or in tandem. The recovery functions must be activated for each database. Activation of recovery features is available to you in the SAS System through the use of the QUEST procedure.

A transaction log, referred to as a KEEPFILE, can be invoked to keep track of every change made to your database. In order to initiate this processing, you must save a database in a special SYSTEM 2000 format. SYSTEM 2000 software provides a full set of commands to invoke the recovery features, such as SAVE, RESTORE, and APPLY.

To invoke recovery using the KEEPFILE recovery vehicle, you must restore a database. Then the updates contained in the KEEPFILE transaction log are applied to the restored database. With this method of recovery, an accidental update to the system can sometimes be nullified. This form of recovery has the least impact on real-time updates but requires greater resources to accomplish the restoration.

A second method of protection is a real-time recovery method referred to as the ROLLBACK feature. The ROLLBACK feature will protect a user from system outages and other similar disasters; it cannot be used to nullify accidental updates as could be done using the KEEPFILE transaction log. Although this method will increase the overhead associated with updates, the ability to recover from system outages without user intervention is many times worth the expense. In addition, SAS procedures that enable updates to your database require ROLLBACK to be activated for a database if you want to guarantee that partial updates do not occur. A partial update occurs when you change data at more than one level of a multi-level database and then make an error. With ROLLBACK disabled, changes to some levels of the database have already been committed and cannot be removed.

SAS procedures that can update SYSTEM 2000 databases via the SYSTEM 2000 engine are FSEDIT, FSVIEW, APPEND, SQL, DBLOAD, QUEST, and "SAS" applications.

Data Security Provided by SYSTEM 2000 Software

Security and administration of data become critical issues when many users are trying to access the same data. SYSTEM 2000 software provides you with the ability to restrict access to your data at an item level. SYSTEM 2000 software provides for many user-defined passwords. Each password can define various combinations of read-and-write access authorities for each group of items related to the password.

Consider a database containing five items named ITEM1 through ITEMS. Suppose ID1 is a password you defined that gave read-only access to ITEM1 and ITEM4 and provided no access to ITEM2, ITEM3, and ITEMS. This situation enables any user using this password to retrieve ITEM1 and ITEM4; an attempt to access any other item results in an error condition from the procedure being used. Attempting to update with read access results in the same error.

362
WHEN SHOULD YOU USE THE SYSTEM 2000 ENGINE?

Any Version 6 SAS application can access data stored in a SYSTEM 2000 database, so why not convert all of your applications to utilize SYSTEM 2000 software? The primary reason is that there is overhead involved with managing data contained in a SYSTEM 2000 database and you should not suffer the cost of that overhead if you don’t stand to gain any benefits. You can translate the previous discussion on the benefits to be obtained by using SYSTEM 2000 software into five areas: real-time shared data access, subsetting of data, redundant or variable length data, sensitive data, and data availability.

Applications That Require Sharing Data
Applications would be easier to maintain and provide more accurate and timely data if the data are maintained in place. Examples of this type of application are inventory systems, scheduling systems, purchasing systems, and project management systems.

If your application requires that changes normally occurring to data be reflected to users when they occur, you will have to provide a means of updating the data while other users are accessing it. Usually, this means you have to queue the use of the data so that you can gain exclusive use. In a large network with many users accessing your data, this is difficult to achieve, and it removes the data from the user for considerable periods. Of course, the SAS/SHARE product, when available in Version 6, will help this situation.

If your application requires that data be accessed by many users during a work period and updated during that same period, you should consider moving your data into a SYSTEM 2000 database and utilizing Multi-User software to access the data in real time. Applications that use SAS/AF software are particularly viable candidates because SAS/AF software affords the most flexible update mechanism of all of the SAS procedures.

Applications That Do Much Subsetting
Consider your company as a hierarchy. The top level of management needs access to all information maintained in the company. As you travel down the hierarchy to lower levels of management, the need to access global information diminishes. When you reach the lowest level, such as an individual or department, very few of the total corporate data need to be presented.

Many applications are designed such that department data are gathered together at least once per day and combined in such a way that the corporate view of the data is updated. Once the updates occur, the data are split back down into their lowest form for access by the departments. Many Version 5 SAS applications utilize this method of maintaining data. Using Version 6 of the SAS System, you can reduce these messy preprocessing and postprocessing steps, if not eliminate them.

The global WHERE option in SAS software enables you to determine which observations will be passed to a procedure, eliminating the need to create a subset of data with a DATA step before executing the procedure. If your data are generally subsetted using more than one index, the SYSTEM 2000 engine can enhance this process and, therefore, makes your application a candidate for utilizing SYSTEM 2000 software.

Applications That Contain Redundant Data
As in all relational data structures, the standard SAS data set many times contains large amounts of redundant, repetitive data. Inventory systems where the department name is carried in the record, cause many unneeded occurrences of that name within the data. If

your data consist of a large one-to-many relationship, you can create a hierarchy of data within your system and eliminate a great deal of redundant data by using SYSTEM 2000 software. Applications that are designed to take advantage of naturally occurring hierarchies can result in significant DASD savings for utilizing SYSTEM 2000 software.

Unneeded trailing blanks are another form of redundant data. All applications must define data fields to be able to hold the maximum field length of data. Normally this results in millions of bytes of wasted DASD space. SYSTEM 2000 software can reduce the amount of wasted space through the use of variable length storage techniques. This provides significant savings over storage required for Version 5 SAS data sets. However, the COMPRESS option in Version 6 of the SAS System can greatly reduce space in many cases beyond savings capable with SYSTEM 2000 software, especially when your data have a low one-to-many ratio.

Applications That Contain Highly Sensitive Data
Banking applications, military applications, industrial research and development applications, and personnel applications are all examples of applications that contain data that are highly sensitive in nature. The normal method of restricting access to this data is to globally secure any data set that contains sensitive information. This can result in many copies of data, each view built to suit the needs of the particular user involved.

SYSTEM 2000 software can simplify this process with use of security features that enable you to secure data at an item level. This means that you only have to maintain one copy of the data and all users will access the same copy. Access to sensitive data will be restricted by password and item level, not at the data set level.

Applications That Must Maintain a High Degree of Availability
It is extremely important for some systems to be able to maintain a high degree of availability. One aspect of a system that can reduce downtime is the system’s ability to recover from errors. Some errors occur through improper use of a system or through user error. Other errors occur as a result of some component’s failure to perform properly, such as a power outage. If your systems have a requirement for data to be available with very little downtime, you need to concern yourself with how to recover from errors.

The built-in error recovery features available with SYSTEM 2000 software will help you reduce time to recover from errors in any system utilizing its database structures. In many cases, recovery from errors can be performed automatically by SYSTEM 2000 software.

HOW DO YOU CONVERT AN APPLICATION?
If you have decided that you want to move an application to SYSTEM 2000, what must you do? Before attempting to actually convert your SAS data into a SYSTEM 2000 database, look carefully at your application and see if the data structures need to be altered.

Reviewing the Design of Your Data Structures
Review all of the SAS data sets that your application uses. When you execute the SAS procedures to define and populate your SYSTEM 2000 database using an existing SAS data set, you will need three pieces of information about each item you intend to maintain in the database. You must determine at what level in the database hierarchy the item is located, its desired length, and whether or not the item should be used as an index.

Determine whether you want to take advantage of the hierarchical data structures used by SYSTEM 2000 software. It is possible to
reduce space requirements by utilizing multiple data levels, but this can complicate your updating processes. In order to obtain space reduction through data redundancy, you must ensure that your update process always appends data to the lowest level record possible. If reduced DASD space requirements are one of your goals, consider this step carefully.

Since your database definition does not need to be the same as your view definition, look for ways in which you can make optimum use of SYSTEM 2000 software does well, while not affecting the end user's view of the data. Identify character fields that can make use of variable length data storage utilized by SYSTEM 2000 software. In general, items that are eight or more characters in length and either contain many trailing blanks or tend to be valued very often should have actual definitions that are smaller than what is required by the application.

Finally, determine which items you will be using for selection criteria. You should select at least one item at each level of your definition that will be indexed. All items can be identified as indexes, but you should restrict items to be indexed to those which you intend to use in selection processes frequently. Because all SAS engines have the ability to select data without the use of indexes, items that are only occasionally used in selection should not be indexed.

Defining Your Database Structures

The next step in your conversion process is to define the database. There are two papers in the SUGI15 proceedings that discuss the DBLOAD procedure and the hierarchical database engine. "A Hierarchical Database Engine for Version 6 of the SAS System," written by Barbara Barrett, discusses the engine that SAS software uses to access SYSTEM 2000 data. "Using the DBLOAD Procedure to Create and Populate SYSTEM 2000 Data Management Software Databases," written by David Pitts, deals in depth with the process of creating and populating databases using SAS data sets as input.

For this paper, it is enough to know that the DBLOAD procedure can be used to create SYSTEM 2000 database definitions and populate the database. Without exercising any options, the DBLOAD procedure will create a default definition based on the existing SAS data set. The hierarchical structure will consist of one level: that is, it will be a flat database and will contain item names and lengths identical to those in the SAS file. If you wish to take advantage of the benefits available by using the unique features of SYSTEM 2000 software you have to modify the default definition created by the DBLOAD procedure.

The DBLOAD procedure will let you change item names, change item lengths, identify items that are to be indexed and identify the level within the hierarchy where the item resides. It will also enable you to exclude items from the input file. All of these definition changes can be implemented online during DBLOAD procedure execution under display manager except changes to item length. If you wish to take advantage of variable length data items, you must specify the S2KLEN statement. Figure 6 shows the contents of the SAS data set INVENT, which contains the data for the INVENTORY system. I will use this data set as an example of how to create a definition with the DBLOAD procedure.

### Table: INVENTORY Data Set Contents

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Pos</th>
<th>Format</th>
<th>Informat</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>SITEPW</td>
<td>Num</td>
<td>8</td>
<td>249</td>
<td>10.</td>
<td>10.</td>
</tr>
<tr>
<td>39</td>
<td>DTITLE1</td>
<td>Char</td>
<td>32</td>
<td>821</td>
<td>$32.</td>
<td>$32.</td>
</tr>
</tbody>
</table>
INVENTORY DATABASE Definition

In Figure 7, you will notice that the record names are NAMEREC, ITEMREC, and DOCUMENTATION. These are the names of the records in an existing SYSTEM 2000 database. When you allow the DBLOAD procedure to create your database, it generates names for records like RECORD_LEVEL_1, RECORD_LEVEL_2, and RECORD_LEVEL_3. You can use the QUEST procedure to rename your records if you desire.

Making Changes to Your Data

Once you have created your database and populated it with existing SAS data, you will need to make some changes in how you maintain the data. Any applications that use the DATA step or the MERGE procedure to combine data will have to be modified because only SAS data files can be created in a DATA step. One suggestion is to create a SAS data file containing the changed observations and then use the DBLOAD or APPEND procedure to add them into your database. Make sure if you are updating existing entries in the fashion that the old version of the observation is deleted from the database.

However, with the extensive improvements in the features and functions available with SAS/AF software, you should use it as the basis of your applications. The SAS/ACCESS interface to SYSTEM 2000 software supports all SQL update and retrieval commands for use by SAS/AF applications. When you design your applications utilizing SAS/AF programs and the SYSTEM 2000 engine, you can allow multiple users concurrent access to SYSTEM 2000 data in either retrieval or update mode.

The FSEDIT and FSVIEW procedures enable you to make updates to your SYSTEM 2000 databases. With Version 6 of the SAS System, the functionality provided by these procedures has been greatly expanded over what was available in Version 5 of the SAS System. As an example, you can specify programs to edit observations that are very similar to SAS/AF applications.

CONCLUSION

Applications that are running smoothly should not be altered. There are very few applications of that nature. If you are experiencing resource shortages, want to improve performance, or would like to enhance availability of your data, you should consider SYSTEM 2000 software as a vehicle to help you implement the changes you desire.

SYSTEM 2000 software can reduce space requirements, provides a Multi-User product to enable concurrent access to data, and has extensive built-in security and recovery features. SYSTEM 2000 software can help improve the overall function of an application.

Finally, Version 6 of the SAS System provides SAS/ACCESS interface to SYSTEM 2000 software, which enables all procedures and the DATA step to access SYSTEM 2000 databases. The FSEDIT, FSVIEW, DBLOAD, APPEND, and SQL procedures as well as SAS/AF applications can update data contained in a SYSTEM 2000 database.

SAS, SYSTEM 2000, SAS/ACCESS, SAS/AF, and SAS/SHARE are registered trademarks of SAS Institute Inc. Multi-User is a trademark of SAS Institute Inc.

IBM is a registered trademark of International Business Machines Corporation. SQL is a trademark of International Business Machines Corporation.