How to Get the Most out of Your SYSTEM 2000® Data Management Software

David W. Pitts, SAS Institute Inc., Austin, Texas
Roy D. Vance, SAS Institute Inc., Austin, Texas

ABSTRACT
The Version 6 architecture has opened up new ways to interface the SAS® System to SYSTEM 2000® software databases. This paper shows you how to take advantage of some of the powerful features of SYSTEM 2000 software. Such features include Coordinated Recovery, audit trail update log, Multi-User™ software, higher level language support through PLEX, security at the individual component level, and SYSTEM 2000's own native Soft-Contained Facility (SCF) languages.

SYSTEM 2000 software can be used in conjunction with Version 6 interfaces or as a stand alone data management software system that provides an easy and convenient means of rapid update and retrieval of data.

INTRODUCTION
SYSTEM 2000 Data Management software provides a way to extend the SAS System's data management, query, security, and recovery capabilities. SYSTEM 2000 software offers fast data access, flexible data query, data item security, and Coordinated Recovery. The SAS System offers the powers of data analysis, presentation capabilities, ease of use, and familiarity. The SAS/ACCESS® interface to SYSTEM 2000 Data Management Software is the product that makes it all possible. The SAS/ACCESS product enables you to describe SYSTEM 2000 data to the SAS System. You store the description in SAS/ACCESS descriptor files, which you can use in any SAS procedure just as you would use SAS data files. This allows you to write SAS programs that take full advantage of the powers of SYSTEM 2000 and at the same time does not require you learn the DBMS's programming concepts. Furthermore, you can use the same SAS program whether your data reside in a SAS data set or in a SYSTEM 2000 database.

SYSTEM 2000 is a hierarchical data management system for mainframe computer systems running under the MVS and CMS operating systems. The hierarchical structure, as used by SYSTEM 2000 software, provides for efficient data storage, enables rapid access for updates and retrievals, and is exceptionally well-suited for specific data qualification and sorting. Security is provided at the database level and for each item within the database through passwords. You can access the database for any function of SYSTEM 2000 through SCF (English-type statements), PLEX (Programming Language Extensions for PL/I, FORTRAN, COBOL, or assembler language programs), and the SAS/ACCESS interface to SYSTEM 2000.

Multi-User software provides an excellent solution for simultaneous access and coordinated updates by multiple users. Multi-User software is ideal for transactional processing by SAS/FSP® and CICS PLEX programs. All of the capabilities of SYSTEM 2000 single user, as well as security at the individual user level, are maintained.

The following sections describe how you can get the most out of SYSTEM 2000:

- Provide for security using DBA and secondary passwords to assign specific authorities at the individual item level.
- Provide for simultaneous update and retrieval access using the Multi-User product.
- Take advantage of execution parameters to tune your Multi-User system.
- Use Coordinated Recovery to ensure database integrity by preventing partial updates.
- Use the SAS/ACCESS interface, SCF through the QUEST procedure, and PLEX facilities to tailor your applications for your needs.

SYSTEM 2000 ESSENTIALS
The following sections discuss the basic concept of SYSTEM 2000 databases, records, items, indexes, where-clauses, and security.

SYSTEM 2000 Databases
SYSTEM 2000 databases are hierarchical in that you store and access data according to organized relationships between groups of associated data. A database is created by specifying the NEW DATA BASE IS command, then using the DEFINE language in SCF to define the schema records and items. The DBLOAD procedure in SAS/ACCESS software can also be used to initially create, define, and load a database. Either way, you create a database definition consisting of schema records and schema items.

SYSTEM 2000 Schema Records
A database consists of groups of logically related data called logical entries. A logical entry contains one or more schema records. The top record is called the entry record and its database level is zero. The first descendant record is at database level one, and so forth up to 32 levels. Each level zero schema record defines a logical entry. In a hierarchy, the relationship between the parent record and a descendant record is one-to-many. Or looking at it the other way, each descendant record has only one parent record. Each schema record contains one or more schema items.

SYSTEM 2000 Schema Item Types
Every schema item in a SYSTEM 2000 database is of a particular item type that specifies how the data values will be stored and displayed. The following are the available SYSTEM 2000 item types:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td>stores whole numbers.</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>stores numbers with a decimal point.</td>
</tr>
<tr>
<td>MONEY</td>
<td>is the same as DECIMAL but provides for a floating dollar sign at the left and CR on the right (if negative) when displayed.</td>
</tr>
<tr>
<td>REAL</td>
<td>stores either single- or double-precision floating point numbers.</td>
</tr>
<tr>
<td>DOUBLE</td>
<td></td>
</tr>
<tr>
<td>DATE</td>
<td>defines a date and is displayed according to the format requested.</td>
</tr>
</tbody>
</table>
CHARACTER stores alphanumeric values with trailing, leading, and extra embedded blanks removed.

TEXT is the same as CHARACTER except the blanks are preserved as entered.

UNDEFINED stores any other type of data, especially binary or hex data.

Character Overflow Feature

A unique feature of SYSTEM 2000 software is character overflow that allows for more efficient data storage. CHARACTER and TEXT fields defined as four or more characters in length can hold up to 250 characters of data, although space in the data table is limited to the defined length. When a data value exceeds the defined length, a pointer in the data table points to the displacement in the overflow table where the data resides. This allows you to define the length of your items that will hold most data values but still accommodate values up to 250 characters when necessary. This technique can save a lot of disk storage if your database contains optional comment fields that are rarely valued.

Your access and view descriptors define the length of a database item. If the length is not altered in the descriptor, the default is the SYSTEM 2000 item definition length. Therefore, to prevent truncation when retrieving values that overflow, the descriptor length must be as big as your largest value. Note that SYSTEM 2000 overflow values can be up to 250 characters; the maximum length you can specify in a descriptor is 200.

SYSTEM 2000 Indexing

SYSTEM 2000 software offers complete and flexible indexing for fast data access. You identify the indexed or key item when you define the database. Any or all items can be defined as a key item. As you input data into your database, the indexes are maintained automatically without affecting the database structure. Ideal candidates for keys are items that are mostly unique, such as social security numbers and account numbers, or items you will most likely use in a where-clause, such as last name, state, and region.

Two master password commands CREATE INDEX and REMOVE INDEX allow you to change index specifications after the database has been defined. These commands can be used to enhance performance. If you plan to initially load your data in increments or add a significant amount of new data, you may speed up the load process by removing the index until all the increments have been inserted. Then create your indexes again after all the data have been loaded.

Selecting a Subset of Data

You can subset the data you want to process by using a SYSTEM 2000 where-clause. Although specifying a key item is the most efficient way, you can specify any item for which you have where-clause authority. If you need to specify a non-key item in a where-clause, it is more efficient if you can add a key item with the AND connector. Then the non-key item search is limited to the records selected by the key item.

You have a wide range of qualifications operators and connectors to select from: unary (EXISTS, FAILS), binary (EQ, NE, GE, GT, LE, LT), ternary (EQ, SPANS), and connector operators (AND, OR, NOT). You can also use their equivalent symbols. CONTAINS matches on a subsetting of characters within a CHARACTER or TEXT item. The HAS operator allows for processing descending and ascendant records, and the AT operator allows you to specify a data record by position. Parentheses can be used in conjunction with any of the above operators for precise processing control.

Multi-User Software

Multi-User software allows more than one user to update or retrieve the same or different data concurrently. Simultaneous access for up to 63 databases and up to 230 users is allowed. Data integrity is ensured because Multi-User software is aware of each record being updated or retrieved and is able to prevent any conflicts by temporarily suspending a conflicting activity. For example, if two requests to update the same record are received simultaneously by Multi-User software, one is dispatched to process the request, and the other would be suspended until the record is freed. The SYSTEM 2000 interface view engine handles this type of conflict slightly differently, but you do not need to be concerned about it. Multi-User software, functioning like a mini operating system, resolves any conflicts and provides the necessary data protection.
In addition to controlling simultaneous access from many users for many databases, the real power of Multi-User software is its ability to multi-thread concurrent processing. As with any program, it is often necessary for a user within Multi-User software to wait for an event to occur, such as I/O completion, before processing can continue. When such a wait occurs control is relinquished to Multi-User software and processing continues on behalf of another user. Multi-User software, therefore, allows the I/O operations of one user to be overlapped by the processing of other users. When a user has to wait for the completion of an I/O operation, Multi-User software sets that activity in a wait state and selects for dispatch a user that is ready to run and has the highest priority. Furthermore, Multi-User software keeps I/O to a minimum by allowing all users access to the same page of data that is in memory. Updated pages are written back to disk only when necessary so the buffer can be reassigned to another page, or in the few instances where it is necessary to have all data cleared to disk.

Multi-User Software Tuning Considerations

Tuning the Multi-User system to best suit your needs is provided through execution parameters. These parameters are discussed in detail in Product Support Manual, Release 11.6, IBM MVS and CMS. Up to 53 threads and 230 concurrent users are allowed, but you probably will find that 8 to 10 threads and 25 to 30 users will be adequate. A thread is a channel for a command currently being processed. With 8 threads there can be 8 commands being processed simultaneously. The USERS parameter specifies the maximum number of users that can be active in a thread or queued waiting for a thread at any given time.

Another significant method to fine-tune your system is the POOL parameter, which allocates the number of buffers Multi-User can use. There are two types of buffers: scratch (work) and database file buffers. Scratch file buffers are associated with a user and database buffers with a database. The size of the scratch file buffer is determined by the smallest POOL specification with a type B (both) or S (scratch only). Data is written to database buffers only when a command is ready to use. It is possible to specify up to 20 different database block and buffer sizes to choose from. This gives you the ability to isolate particular database files (usually the index, record, or data file) by giving them a unique block size and a POOL of type DE (database and exact fit only). While there is no set rule for determining the number of buffers needed, 4 work buffers for each user and 10 data buffers for each database is a good starting allocation. After you are running in a production environment, you can use the Multi-User tuning tools to determine your I/O activity.

To further aid in tuning your system, Multi-User software provides for an Accounting Log and a Diagnostic Log. SAS Technical Report S2-106, Multi-User Tuning Tools for SYSTEM 2000 Software, Release 11.6 under IBM MVS and CMS offers additional Multi-User console commands and sample SAS programs to analyze the two log files.

Some advantages of using the Multi-User system are:

- one copy of SYSTEM 2000 software code for all users
- data integrity ensured by Multi-User software
- data redundancy and overhead of separate files (data and work) reduced
- resources between users (buffers, threads, data) shared
- all users allowed simultaneous access to current data.

SYSTEM 2000 SOFTWARE BACKUP AND RECOVERY

Each data center has a basic backup system in case of catastrophes where the disk storage devices are no longer usable. In addition to backup systems, some protection is needed for accidental erasure of data, program logic errors, and abending transactions. SYSTEM 2000 software provides for both types of recovery. The first one is the basic recovery that provides for a backup copy of the database. The second one, called Coordinated Recovery, is a process that SYSTEM 2000 software automatically uses to ensure the integrity of one or more databases.

Basic Recovery

Basic recovery provides a complete backup copy of the database and optionally allows for an update log. It is strongly recommended that each database be periodically backed-up by simply issuing the SAVE database command. The backup copy of the database can be brought back at anytime using the RESTORE <database-name> command.

Optionally, you can request that an update log be maintained. You can enable the update log with either the SAVE or RESTORE command. Once you have enabled the update log, pertinent internal information is collected for each update cycle. Periodically you need to save the update log by issuing the KEEP command to move the data from the update log (database file 7) to the keepfile (usually a tape or another disk file). Should recovery become necessary, you need to issue a final KEEP command to copy the remaining data on the update log to the keepfile, then RESTORE the database. When the restore is complete, you can APPLY ALL or APPLY THRU <cycle-number> to recover the database up to the point where it became damaged. Each of these commands must be submitted by you.

Coordinated Recovery

The other mechanism is Coordinated Recovery. This process is automatically invoked when necessary, thereby preventing a loss of downtime for recovery, and also preventing errors that can be introduced with manual intervention. This process eliminates most headaches in recovering damaged databases. It is totally automated, and it is reliable. When an error condition occurs or a damaged database is opened, the recovery process is automatically invoked. In many cases you will not even know it happened, unless you examine the job log.

Updating a database using the SAS System takes advantages of Coordinated Recovery if it is enabled. If you have Coordinated Recovery enabled, there will be no partial updates caused by database error, transaction abort, or the CANCEL command.

Coordinated Recovery offers both timely and automatic support and does it efficiently. Some events such as a power failure or a Multi-User abort need not have a disastrous effect; they can be reduced to an inconvenience by employing Coordinated Recovery. You do not have to intervene in any way. Only incomplete updates are removed; all other committed updates will be in the database and readily available for use.
SYSTEM 2000 PROGRAMMING LANGUAGE EXTENSIONS (PLEX)

With PLEX you combine the general capabilities of a programming language such as looping, conditional branching, flexible syntax, arithmetic operations, and data editing with SYSTEM 2000 software's data retrieval, update, and maintenance capabilities to serve a wide variety of applications. PLEX programs can be executed in single-user or Multi-User and in batch or TP environments such as CICS and TSO. PLEX gives your high-level language applications complete control of SYSTEM 2000 databases. Where the Self-Contained Facility (SCF) is limited to one database, PLEX has no limitations. Up to 63 databases can be opened, 15 locate files and stacks are available, and with Coordinated Recovery, complete control is available to coordinate the updating of all databases.

An important part of coordinating recovery of two or more databases is the ROLLBACK command. This command, available only in PLEX, causes recovery to be invoked only for your updates. Its purpose is to help you synchronize updates to one or more databases. For example, you may want to modify a record in database A and insert a record in database B. If the update to database B fails for any reason, you want the update to database A to be canceled. To accomplish this, you simply issue the ROLLBACK command. The database is rolled back and then forward. The incomplete update to database A is omitted during roll forward. When both databases have been successfully updated, you can issue the COMMIT command to permanently mark your database updates as accepted and to drop the hold on those records so that other Multi-User applications can now update them.

SELF-CONTAINED FACILITY (SCF)

SCF comprises five varieties of English-like language for data updates, retrievals, definition, and control. These languages are designed for the user with little or no programming skills. The languages are easily learned and include complete diagnostic messages. All SCF commands can be used in either an interactive or batch environment.

If you are using the SAS system, the QUEST procedure allows you to access a SYSTEM 2000 database directly without using a view descriptor for flexible ad hoc browsing and updating. You have full access to all SCF languages either in interactive display manager, interactive line, or batch modes. Both Multi-User and single-user environments are supported with only minor differences. You can use the power of SAS/AF software or SAS macros to provide frontend parsing of input data and generate the desired SCF commands to meet your needs.

SCF is ideal for browsing through the database with ad hoc queries. With the QUEST language you have the PRINT, TALLY, and LIST commands. PRINT and LIST can use the ORDERED BY and WHERE clauses. The PRINT command displays values in a simple, sequential list. The TALLY command exclusively uses the index tables to display information about key items. It can give the number of times each distinct value occurs, the total number of distinct values, and a count of all the values stored for the item.

The LIST command is the most flexible retrieval command in that it displays data in a columnar format and specifies report title, column headings, and a footing line. You can also store a LIST command as a SYSTEM 2000 string to produce periodical reports easily. The following are two sample LIST commands and output showing how to use SCF:

**Output 1 Sample LIST Command Output**

```
LIST C1, LAST NAME, OB C2 WHERE C1 <= 1005;
```

---

**Output 2 Sample LIST Command Output**

```
LIST /TITLE ('55) D(15)REPORT TITLE, F(15) "FOOTING LINE " ---,
1(10) LAST NAME, EMPLOYEE NUMBER, F(15) SOCIAL SECURITY NUMBER,
C1, C2, C3, OB C2 WHERE C1 <= 1005;
```

---

There are five basic command languages for you to use:

- **DEFINE** allows you to define and redefine database schema records and items, stored strings, and functions.
- **CONTROL** allows you to control administrative functions, such as saving and restoring a database, assigning passwords and authorities, creating/removing indexes, and enabling features like Coordinated Recovery.
- **ACCESS** or **QUEST** allows you to access the data for retrieval and updates.
- **QUEUE** allows you to group your updates and retrieval commands into batches to process large runs efficiently.
- **REPORT** allows you to produce specialized reports.

**Strings and Functions**

Commands can be partially or completely contained in a string definition. A string is a component in the database definition and as such is assigned an item number and an item name, either of which can...
be used to invoke the string. A string can contain an SCF command, part of a command, or a series of commands. Strings provide a means of storing often-used commands and can be either 'simple' or parametric. A parametric string has positional parameters, which are replaced with values when the string is invoked. The values replace the positional parameters specified at definition time. A common use of strings is to produce recurring reports using the LIST command.

Functions allow you to store arithmetic expressions and to specify a numeric type for the display of the output. Along with the user defined functions, SYSTEM 2000 software provides the system functions COUNT, SUM, MAX, MIN, AVG, and SIGMA.

CONCLUSION

SYSTEM 2000 Data Management System software offers the storing of data that allows for fast data access, flexible data query, full security at the schema item level, automatic Coordinated Recovery, Multi-User software, PLEX, and the Self-Contained Facility. The combination of the Multi-User product and the Coordinated Recovery feature offers an excellent solution when the application demands simultaneous updates and retrievals in an online transaction environment.

Beginning with Version 6 of the SAS System, the SAS/ACCESS interface to SYSTEM 2000 software allows any SAS procedure to use a view of a database just as you would use a SAS data set. This gives the SAS user the full advantages of SYSTEM 2000 software and still allows you to use the SAS System tools that are familiar to you.

SAS, SAS/ACCESS, SAS/AF, SAS/FSP, and SYSTEM 2000 are registered trademarks and Multi-User is a trademark of SAS Institute Inc., Cary, NC, U.S.A.

IBM is a registered trademark of International Business Machines Corporation.