DISK INVENTORY AND CONTROL SYSTEM:
A REXX TOOL FOR THE MANAGEMENT
OF SAS® DATA BASES UNDER CMS

Arthur L. Carpenter
California Occidental Consultants

Key Words
CMS, REXX, Interface, Data Management,
Inventory

Introduction
Imagine a large data collection project
where several types of data have been
collected over a number of years in as
many as a dozen different task areas.
Further complicate these conditions by
using multiple subcontractors and by
changing them in the midst of the project.
Add spice by changing the sampling and
project objectives during the course of
the study. Implement the data entry and
management using poorly trained and
constantly changing personnel. And create
over a hundred and eighty data bases using
widely varying data collection techniques
and consisting of a variety of internal
structures. Add to this growing nightmare
a low budget (naturally), a need for total
data traceability and final SAS data bases
suitable for use in litigation.

Using this prescription for disaster as a
challenge, various data management
concepts are discussed and used as the
motivation for the design of a Disk
Inventory and Control System which inter­
faces the CMS operating system with SAS
using REXX. Problems discussed include
multiple user update access to a data
base, data base establishment software
control, selected user access, the tracing
of raw data updates, simplified non-CMS
commands, and data base documentation and
management tools.

The discussion includes the application of
the control system to a marine biological
monitoring study and provides suggestions
and insights for the implementation at
other sites.

Creationism versus Darwinism
Large data collection and management tasks
provide some unique opportunities for
those responsible for the maintenance and
documentation of the data bases. In large
studies, the power of SAS becomes both a
blessing and a challenge by providing
individual users with the ability to
access and modify information with little
or no direction from the controlling
organization. Yet, the same power
that frees up programming resources may
also cause problems with the control and
documentation of the data bases being
maintained.

Often, the responsibility of the
maintenance of the data base(s) cannot be
left to the individual programmers or
users. Many times individuals lack the
training and/or motivation to take the
time and effort necessary to keep the data
bases from falling into disrepair. The
result is a tangle of data base threads
that, like spaghetti, become increasingly
convoluted as they evolve and change.

Generally speaking, data bases are either
designed before being created or they
evolve without (or sometimes in spite of)
a prior plan. Planned data bases have a
number of obvious advantages, such as
control and documentation, however this
control is achieved at the expense of
increased rigidity. Data bases that are
not particularly planned, on the other
hand, are at times more flexible, but
often impossible to control.

Data bases that evolve often start small
and are easily managed by a single user.
Over time (sometimes a short time) they
grow, become more complex, and lose their
individual identities as they are merged
with other related data bases. As the
number of data base users also grows, so
do the problems associated with the
documentation and maintenance of the data
base(s). It becomes increasingly
difficult to establish or maintain naming
conventions, establishment protocols, and
modification tracking.

Rigid or even written guidelines rarely
exist for evolving data bases. Specific
requirements for documentation and
configuration control are usually left to
the discretion of the programmer or user.
Certainly this seems to be much more
flexible and flexibility has intuitive
appeal to most programmers (the
flexibility of SAS is one of its
hallmarks). However, if too much
discretion is applied, the evolving data
base can become so "flexible" as to be
useless.

Large data bases do not need to be a mess
and although some flexibility is lost when
data bases are managed, they don't need to
be rigid. Data bases can be created using
a definite plan and specific objectives.
Data bases that are planned and then
created according to that plan are less
likely to have the problems associated
with unplanned and evolving data bases.
The trade-off is between control and
flexibility. Data bases that are planned
and maintained accordingly require the
exercise of control. Part of the planning
process is to define the objectives of the data base and then configure the data base to meet those objectives.

Data Base Objectives

Data bases are created for a purpose and this purpose drives much of the form of what the data base is. As obvious as this sounds, the purpose of the data base is often ignored when it comes time to create it. Even temporary WORK data sets have a purpose. PROC CORR requires a data set with the values of each variable arranged with its own variable name. PROC GLM, on the other hand, requires these values to be within a single variable and to be associated with a CLASSification variable (Table 1). The purpose of the data set (CORR vs. GLM) determines its form.

There are also objectives and purposes associated with larger systems of data bases. There should always be more factors under consideration when a data base's objectives are established than just the anticipated types of analysis. Often objectives of a data base include user availability, documentation of the data, ease of use, security, ease of updating, standardizations, etc.

Table 1

<table>
<thead>
<tr>
<th>Data Set Used By PROC CORR</th>
<th>REP</th>
<th>ARSENIC</th>
<th>CADMIUM</th>
<th>MERCURY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>5.2</td>
<td>4.4</td>
<td>2.4</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3.8</td>
<td>3.2</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Set Used By PROC GLM</th>
<th>REP</th>
<th>METAL</th>
<th>MGPERKG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>ARSENIC</td>
<td>5.2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>CADMIUM</td>
<td>4.4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>MERCURY</td>
<td>2.4</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>ARSENIC</td>
<td>3.8</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>CADMIUM</td>
<td>3.2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>MERCURY</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Configuration Control

The configuration of the data base is the implementation of its primary objective or purpose. All data bases regardless of size, should meet some minimum guidelines. These guidelines or data standards will of course vary from project to project, but most will have some items in common.

- The data base should be constructed to meet its primary objectives.
- The data base administrator should be able to exercise whatever control is necessary to maintain the objectives and integrity of the data base.
- The final data base should be traceable to its original source.
- The data base should be protected from accidental updates and/or modifications.
- The data base should be designed for maximum utilization of computer resources.
- Standards for data entry, modifications and updates should be established.
- Naming conventions, formats, documentation standards should be established.
- Procedures for the security of the data base, whether from unauthorized use or modification, should be established.

Prescription for a Disaster

Without configuration control the data base developer is asking for trouble. There are a number of common excuses (reasons) for not needing controls for some particular project. Table 2 lists a few of the more common of these.

Table 2

<table>
<thead>
<tr>
<th>PRESCRIPTION FOR A DISASTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>This Project is Short Term</td>
</tr>
<tr>
<td>This is a Low Budget Project (&quot;We'll do it quick and dirty.&quot;).</td>
</tr>
<tr>
<td>One Person Will Handle All Aspects of the Data Base.</td>
</tr>
<tr>
<td>Data Entry is Easy (&quot;Who needs training?&quot;).</td>
</tr>
<tr>
<td>Data Entry is Easy (&quot;Quality control checks are not needed.&quot;).</td>
</tr>
<tr>
<td>Standardization is not needed (&quot;Data will not be shared.&quot;).</td>
</tr>
<tr>
<td>Documentation will be done after this fire drill (&quot;We'll do the documentation as soon as we have time.&quot;).</td>
</tr>
</tbody>
</table>

A short term study of kelp plants and water clarity was initiated in the waters off of Southern California in 1976. Over the next six years a number of additional topics of interest were added to this now ongoing study. The data were kept in raw files and were analyzed and summarized using various FORTRAN programs. By 1983 as many as eleven subcontractors were collecting and analyzing their own data, and each subcontractor was responsible for several individual studies, each of which resulted in one or more data files.
The objectives and even the direction of individual studies was constantly changing. Principal investigators changed and those that didn’t change often changed the emphasis of their studies. During these transitions, early undocumented data was subject to being lost or misplaced and poorly trained data technicians were not always consistent with the handling of the data.

By 1983, it was apparent that the amounts of data being collected and the fact that the data was to be used in litigation, warranted the implementation of specific data controls. SAS was installed (on an IBM 4331 under CMS) in order to consolidate and standardize the growing number of data sets. However, SAS alone could not meet the configuration objectives stated above.

**Disk Inventory and Control System**

A management tool known as the Disk Inventory and Control System, DICS, was written using REXX in order to provide the control the project required. The system is simply a series of batch processors that maintain control of the data bases by performing the data base updates (Table 3) and saving the results on a set of predefined mini-disks. An inventory of all data base changes, archives, retrievals, etc. is also maintained.

**Table 3**

<table>
<thead>
<tr>
<th>OVERVIEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disconnecting Machines that Process All Requests From Authorized Users.</td>
</tr>
<tr>
<td>One Main Controlling Machine (BATCH OPERATOR) Acts as a Traffic Cop.</td>
</tr>
<tr>
<td>Other Machines (BATCH FACILITIES) Perform All Data Processing Activity.</td>
</tr>
<tr>
<td>All Processing is Logged for Documentation and Status Purposes.</td>
</tr>
<tr>
<td>All General Data is Maintained Under the Control of the DICS. Users have Read Access.</td>
</tr>
<tr>
<td>Interfacing With Existing Archive and Inventory Systems.</td>
</tr>
</tbody>
</table>

Prior to the implementation of centralized controls, each subcontractor was responsible for the design and maintenance of each data base containing data collected by that subcontractor. Individual subcontractors did not always have trained data processing personnel and the resulting SAS data bases often lacked standards and controls. The result was a series of data bases with little in common even within the purvey of a subcontractor.

Since the motivating study utilized a number of contractors each with a variety of data bases and processing needs, the DICS was designed to control one or more separate projects at any given time. Each project or subcontractor is assigned a set of five predefined mini-disks. Write access to four of these disks is controlled by DICS (Figure 1). These mini-disks and their assigned file modes are:

- B SOFTWARE
- C RAW DATA
- D SAS DATA BASES
- E REPORT / GUEST DATA
- F CONTRACTOR WORK

**Figure 1**

**CONTROLLING DATA DISKS**

**TRADITIONAL METHOD**

Prior to the implementation of centralized controls, each subcontractor was responsible for the design and maintenance of each data base containing data collected by the subcontractor. Individual subcontractors did not always have trained data processing personnel and the resulting SAS data bases often lacked standards and controls. The result was a series of data bases with little in common even within the purvey of a subcontractor.

Since the motivating study utilized a number of contractors each with a variety of data bases and processing needs, the DICS was designed to control one or more separate projects at any given time. Each project or subcontractor is assigned a set of five predefined mini-disks. Write access to four of these disks is controlled by DICS (Figure 1). These mini-disks and their assigned file modes are:

- B SOFTWARE
- C RAW DATA
- D SAS DATA BASES
- E REPORT / GUEST DATA
- F CONTRACTOR WORK

**Figure 1**

**DATA INVENTORY AND CONTROL METHOD**

The software disk (B) is used to maintain all SAS (and even non-SAS) establishment software used to convert the raw data files into SAS data bases. DICS maintains control over this disk and only the data base administrator may modify the establishment software. This control ensures that only the proper data establishment software will be used during the establishment process. Any modifications to the software are automatically logged by DICS after verifying the submitter’s authority.
The raw data used to create the SAS data bases is placed on the raw data disk (C) at the same time as the new data base is written to the data base disk (D). The raw data and its associated data base are linked automatically and changes in the raw data automatically result in changes to the SAS data base using the establishment programs maintained on the software disk.

The fourth disk, report disk, (E) is used to save SAS programs, intermediate data bases, etc. that need to be documented or tracked for reporting purposes. The facility also exists for a user to grab and hold-on-to a current version of a data base (so results of an analysis will match the data) regardless of changes to that data base made by other users.

The fifth disk is used to maintain non-DICS controlled data, data bases, and programs deemed important by individual users. DICS does not maintain any control over this disk except to ensure that only authorized users have access.

Any event that happens through DICS (a data base update, a software update, an archive or retrieval, etc.) is logged in a file maintained by DICS on one of the predefined disks. DICS uses these logs to monitor the status of all the data bases, and these logs are consulted prior to taking action on user requests e.g. a file cannot be erased unless it was first archived.

Control is maintained by DICS through the use of a series of batch operators and batch facilities (figure 2). The primary batch operator acts as a traffic cop by passing user requests to the batch facilities (Tables 4 & 5). One batch facility is assigned to each project or task area with the batch operator coordinating the activities of the batch facilities. Within each project or task area the batch facility executes requests sequentially, thereby protecting the users from simultaneous file updates by multiple users.

The functions of the batch facilities (Figure 3) include the update of the data bases, establishment software maintenance, report disk updates, raw data and data base erasure, data base data deletions (data modifications), raw data and data base retrieval (and archives). In each case the request is initiated by the user and passed to the primary batch operator. The appropriate project batch facility is then initiated as in Figure 2. The project batch facility then executes the code appropriate to the user's request.
The individual user sees very little of the DIeS system. A menu system exists, however it is not usually required. Because of the control imposed by the DIeS, file naming conventions must be used. Consequently when a user wishes to update a data base, the same code (PS - for Project Save) is used to update all data bases. DICS determines the project and data base(s) to be updated by reading the name of the raw data file to be used. The user need know nothing about software names or execution parameters.

Reasons Why Data Bases Created Using DIeS Are Better

The Disk Inventory and Control System is a complete configuration control system for all of the SAS data bases that it controls and there are six major reasons (Table 6) that data bases established using DIeS are superior to those created directly with SAS.

Table 6
BENEFITS

- Traceability Between Raw Data and Data Bases.
- Standards and Conventions Must be Observed.
- Efficient Utilization of Resources.
- Controlled Processing Using Only Baselined Software.
- Security Against Unauthorized Processing.
- Logs for Documentation and Prevention of Inappropriate Actions.

1) Traceability

Complete traceability is provided in the DIeS. The raw data files that are used to create the data bases are saved on their own disk and are, therefore, always 'baselined'. The software is also always baselined since it is saved on its own disk and only that baselined software is ever used during the establishment process. Logs exist that provide information such as who performed the updates and when the update was performed (Table 7). These logs are also used as control parameters to ensure that files cannot be erased before they are archived and data bases cannot be updated unless the baselined version is retrieved from tape (if that is where it is currently located).

Table 7
Sample Data Base Log

<table>
<thead>
<tr>
<th>File</th>
<th>Date</th>
<th>Time</th>
<th>Action</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>KK01</td>
<td>06/18/89</td>
<td>10:28:15</td>
<td>UPDATE</td>
<td>SMITH</td>
</tr>
<tr>
<td>KK02</td>
<td>06/20/89</td>
<td>09:45:30</td>
<td>UPDATE</td>
<td>JONES</td>
</tr>
<tr>
<td>KK01</td>
<td>07/01/89</td>
<td>13:15:12</td>
<td>DELETE</td>
<td>JONES</td>
</tr>
<tr>
<td>KK01</td>
<td>07/02/89</td>
<td>14:18:32</td>
<td>UPDATE</td>
<td>SMITH</td>
</tr>
<tr>
<td>KK02</td>
<td>07/15/89</td>
<td>15:16:14</td>
<td>ERASE</td>
<td>JONES</td>
</tr>
</tbody>
</table>

2) Precise and Modular Updates

The usual method of update simply merges the new data on top of the old data in a data base. This often obscures the exact source (raw file) for particular observations or values. The DIeS method requires that every data base contain a key variable(s) that defines, for each observation, the raw file that generated the observation. Under DIeS, the new data is not 'MERGED' or 'UPDATED' over the top of the old data. Instead, the old data is completely deleted from the data base and the new data is inserted. This process is extremely valuable in cases where the data must be removed completely and not replaced. To perform this process without DIeS requires the development of specialized software which could easily introduce errors into the data base.

3) Better Utilization of Resources

Since the DIeS is a series of batch processors, resource utilization is improved in several ways. For one, the user simply submits his/her job(s) and then goes on to other tasks. Another improvement is system throughput. Since there is only one batch processor per project, only one establishment may be executing for any given project at any one time. Not only does this ensure equality for computer resources among projects, throughput is increased since multiple jobs for any project must be run serially.

4) Control

Control is one of the best features of DIeS. Users cannot update a data base without using the baselined software
controlled by the data base administrator. Authorization to update a data base can be granted by the data base administrator on an individual basis.

5) Security

Security is enhanced in that users no longer can directly change, update, or erase data bases. Only selected individuals have the authority to use DICS to perform these operations and the lists of these individuals are maintained by DICS.

6) Standardization

File naming conventions are established by the data base administrator and checked by DICS. In addition, the interface to DICS is the same for all users on all projects. The same command is used to establish all data bases. Individuals do not need to know the name of establishment routines or even the data base that is to be created. By using the naming conventions, DICS determines what establishment software to utilize.

Potential Users of the DICS

The control of data bases should be of direct interest to anyone who establishes or maintains data bases of any size or complexity. The use of a formal system like the DICS is of particular value to administrators of projects with multiple users, centralized software houses, users that require data security and control with a minimum or restrictions, and anyone who wishes to have data and software organized and documented.

Summary

The Disk Inventory and Control System consists of only slightly more than 7,000 lines of REXX code that simply, as its name suggests, inventories and maintains control of the data base establishment process. It is user friendly in that less than a dozen commands are needed by the user. Data may be created using any available language e.g., FORTRAN, SAS, dBASE, etc., and the individual user is only required to know the name of the raw data file.

Acknowledgements

The author would like to express his appreciation to the developers of the Disk Inventory and Control System for access to programmer's notes, as well as, developmental information and documentation.

The DICS was designed and created by Kristy Kozicki and Joe Marandola with assistance and input by Helen Shrake, Layne Leach, and Teresa Podimatis. Funding was provided by the Marine Review Committee, Santa Barbara, CA.

About the Author

Arthur L. Carpenter has over twelve years of experience as a statistician and data analyst and has served as a senior consultant with California Occidental Consultants since 1983. His publications list includes a number of papers and posters presented at SUGI and he has developed and presented several courses and seminars on statistics and SAS programming.

Author

Arthur L. Carpenter
California Occidental Consultants
4239 Serena Avenue
Oceanside, CA 92056-5018
(619) 724-8579

SAS is a registered trademark of SAS Institute Inc., Cary, NC, USA.