ALMA: A LIBRARY MANAGEMENT INFORMATION SYSTEM USING SAS/AF© AND SAS/FSP® ON A
DOS/NOVELL® PC NETWORK

by Carmen Vincent and Linda Gowen
Westat, Inc., Rockville, MD

ABSTRACT

This paper describes ALMA, an information system we developed to manage a departmental reference and lending library at Westat. ALMA is programmed in the SAS/AF® and SAS/FSP® systems and runs under DOS 3.31 in a PC network environment under Novell NetWare 386. When we first began to develop the library system, we did not know how to use the SAS/AF® and SAS/FSP® systems and the PC network was in a highly experimental test phase, serving fewer than 100 of the 500 planned users. Our decision to implement the system using these products was a way of learning about the SAS/AF® and SAS/FSP® systems, and SAS® software in a network environment. The ALMA system was also developed to be a demonstration program. This paper describes the design and functions of the library system, and presents some of the teaching materials we developed for presenting the system and the techniques used to create it. We have also included a description of the network, files, and hardware environment needed by the system.

INTRODUCTION

ALMA is an automated system for managing Westat's data processing department's library. The key features of the ALMA system are that it tracks several collections of materials, and coordinates and provides computer assistance for all clerical library activities in one integrated system. ALMA is written in SAS/AF® and SAS/FSP® software, using SCL (source control language). It runs on workstations using DOS 3.31 on a PC local area network running under Novell NetWare 386®.

The Objectives of ALMA

We had three specific objectives for the ALMA system. In addition to our primary goal, to create a system to manage the Westat Computing Systems Library, our secondary goals were to introduce the SAS/AF® and SAS/FSP® systems to ourselves and other programmers at Westat, and to experiment with SAS software and data files in a network environment.

Westat programmers have used SAS software extensively in the past on VAX and IBM mainframes. Recently the trend has been to move many applications to PC microcomputers. At the same time new interactive uses for PC applications are being designed. The SAS/AF® and SAS/FSP® systems have many possibilities for use in management and control systems, and for developing the user interfaces for procedural systems to be used by non-experts. We wanted to test SAS/AF® and SAS/FSP® to see whether they would be useful for Westat in developing these systems.

The Westat Computing Systems Library

The Westat Computing Systems Library is a small, departmental library consisting of several collections of information resources. The Books and Manuals collection includes approximately 650 volumes which circulate and 800 volumes which are on "permanent loan". Books on permanent loan are generally software documentation manuals which have been signed out to programmers for desktop reference materials. In addition, the library has a collection of non-circulating reference materials consisting primarily of software and hardware manuals. A third collection consists of approximately 75 subscriptions to periodicals. There are two more collections which are being added to the library, a collection of databases on magnetic tapes, diskettes, and CD-ROMs, and a collection of code examples and function libraries.

The library serves all Westat employees, but most patrons are staff of the Systems and Applications Programming department. The department includes about 175 programmers, systems analysts, and other technical staff. Those staff are located in six buildings in three cities in two states.

Until last year, a simple dBASE logging system was used to record books checked out from and returned to the library's circulating book collection. The database was a single flat file residing on the PC of a departmental secretary who was responsible for posting the loan and return transactions. There were several unsatisfactory features to this system:

- Records were kept by book title rather than by book, making it difficult to track the whereabouts of individual books when there were multiple copies of one title.

- The database was not readily available either for analysis or browsing.

- It was difficult to attach attributes to the file for related entities such as borrowers.

- Many routine library activities, such as the ordering of new books and keeping track of requests, had to be managed with manual (paper) logs, since they were not supported under this system.

- And there were several collections, including the periodicals collection and the software documentation, which were not a part of the database. These also were managed with paper logs.

DESIGNING THE LIBRARY MANAGEMENT SYSTEM

Our requirements for a library management system were that it be an integrated system for managing all major library activities. We also wanted the system to be expandable, so that new functions and new collections could be added without requiring a redesign of the system. Finally, we needed to broaden the definition of the users of the library system and of its databases.

The Library Collections

Rather than restricting the scope of the library information system to the book collection, the new system includes management information subsystems for other existing and new library collections. Each collection is treated as a separate database with its own special rules (e.g., for lending) and peculiarities (e.g., periodicals are kept for only two years), but is integrated as a subsystem operating within the general library system, using general library functions were possible. This design strategy has made it possible to add several collections, such as the periodicals collection, to the scope of the system after the original design was completed. We believe that this modular, under-one-roof approach will make all collections more accessible to patrons, compared to a one-collection-per-system approach.
The Library System Functions

The functional requirements of the system are best described in relation to the needs of its users. Library patrons, for example, need to be able to browse and search library collections, but do not need to write to the databases or generate reports. In addition to searching and browsing the collections, library staff need to:

- Record orders for new acquisitions, and track the orders
- Place reservations on ordered or checked out materials for patrons requesting the items
- Log in received acquisitions, including donated items as well as ordered or purchased items
- Post loans and returns
- Maintain a patron database
- Maintain a status code for each item in each collection
- Prepare reports such as the catalog lists of the collections, sorted by subject, title, and author
- Prepare reports on various subsets of collections, such as lists of recent acquisitions, lists of books on order, summaries of the number of volumes on loan

Managers responsible for the library need reports to account for library resources, and to monitor the use of the collections. Certain reports are standard, such as reports giving counts of books by type and status, while other reports may be ad hoc (e.g., a list of all patrons at the WestBrook office building who have checked out a copy of a particular manual.)

The Decision to Write ALMA in SAS

We chose SAS as the language of implementation for ALMA for several reasons. The SAS language is flexible and versatile, allowing us to build general purpose modules (such as the module that links a patron’s identifier on a collection record with information on a patron record). And the SAS system has generalized procedures we needed for browsing, searching, and editing.

Another favorable factor is the compatibility of a SAS system with other corporate programming activities. SAS is a frequently used and accepted language of development at Westat for analytical and statistical programs. Nevertheless, programming staff were not familiar with using the SAS/AF and SAS/FSP systems. Therefore, working with those products provided an opportunity for us to learn about the systems, as well as to create a working demonstration of the products to share with other Systems staff.

Why Use the PC Network?

The selection of Westat’s corporation-wide PC network as an operating environment was based on the need to share the collection databases among library staff as well as with patrons. When we made this selection, we had the following options available:

- The Westat DEC VAX system, which could easily accommodate the technical requirements of the library information system. However, not all potential patrons have access to a VAX account, and a general guest account for library users would compromise the security of the system.
- A shared PC in a central area, which could be used by both patrons and library staff. However, the arrangement would be inconvenient for both library staff and for patrons, particularly since our staff is dispersed among six buildings in three cities.
- The Westat PC local area network (LAN), which was in an early test phase of installation at the time we were developing the system. Its disadvantages included its shakedown instabilities and its conservative installation schedule. However its promise was to connect each Westat desktop PC as a workstation on the network, giving access to library files and software for all Systems staff and most analytical staff.

We selected the network as presenting the most advantages. We expected the library catalog files would be more accessible, and that we would benefit from network maintenance and file security. At the same time, we planned to test these network features, and to test the use of SAS software in a network environment.

HOW THE ALMA SYSTEM WORKS

The ALMA system has a menu-driven user interface with menus created in SAS/AF, using the BUILD function. Data are collected chiefly using the SAS/FSP function FSORT, with entry screens formatted and programmed in Screen Control Language (SCL). Browsing and searching in the databases is done through a user interface to FSBRUSE. Preprogrammed reports are produced according to menu options selected by the user, and executed in PROC PRINT, PROC PRINTT, PROC FORMS, PROC FORMAT, and PROC SORT coded in SAS/AF and SCL. Since ALMA’s collection files are SAS database, analyses not included in the report menus can be performed easily using ad hoc SAS procedures and programs.

The Menu Structure

The structure of the ALMA menu system and program modules is illustrated in Figures 1 and 2. The first menu, described in Figure 1, is the collection selection menu. Each collection is a separate SAS data set, and each has a similar set of submenus and programs. As an example, Figure 2 illustrates the complete program and menu structure of the Books and Manuals subsystem.

The first menu of the Books and Manuals subsystem (the second menu the user will encounter) is an access selection screen. The “Access Library” option opens access to individual records in the selected collection. The “New Book” and “Order Book” options are for creating and filling new records in the data set. The “Report Generation” and “Print Catalog” options provide file access rather than record access. Menus and programs available following an Access option include modules for most of the standard library activities, including locating, renewing, returning, requesting, logging acquisitions, and updating status. The data entry screen and annotated program code for the Order Book option is displayed in Figure 3.
The Collection Databases

The collection databases contain one record per item (e.g., book, manual, disk), with each record identified by a unique "call number" which is automatically assigned by the programs for posting orders and logging in donations. The item record contains the descriptive information unique to the item, much like a card in a library card catalog. The information about a new item is entered on a PROC FSEEDIT data entry screen such as the Books and Manuals order screen illustrated in Figure 3. The data elements entered by the librarian for an ordered or donated book or manual include:

- the title(s),
- the author(s),
- a shelving code,
- the publisher,
- the publication date,
- the code of the patron who will be the first borrower, and
- codes of any patrons who have also requested the item.

Circulation status and the order, loan and acquisition dates are automatically filled by the program modules associated with those activities.

Related Databases

To eliminate duplication of patron information on data records, we created a patron database. Patrons are linked to the items they have checked out by their identifying code. The code is the key to the patron database record which includes:

- The patron's name,
- The patron's address (building and room number),
- The patron's telephone number (building trunk code and extension), and
- The code of the patron's area supervisor.

Referencing the patron in the collection databases by code reduces data entry for the library staff, reduces the size of the collection databases, and reduces the effort required to maintain up to date patron information.

We are now considering converting to the use of a publisher's database, using a publisher's code in the collections databases, since our acquisitions tend to be from a fairly short list of publishers. We will then be able to add information about publishers (such as addresses and telephone numbers) while reducing the size of the collection data records.

TRAINING MATERIALS

The documentation format used in Figure 3 was developed to present the ALMA system as an example program when teaching the SAS/AF and SAS/FSP systems to other staff. The curriculum consists of a demonstration of the ALMA system on a PC connected to an overhead projector monitor, accompanied by a discussion of the program documentation, with particular reference to the calling sequences and the use of particular functions within the modules.

THE NETWORK EXPERIENCE

The ALMA library system uses the SAS system loaded as sharable network software on a Novell PC network. The workstations used to program and run the system are 286 and 386 PCs operating under DOS 3.31 locally, and Novell NetWare 386 as network workstations. Because this is a network-based application, the network software must be loaded in the workstation's memory in addition to DOS, leaving less total memory available for SAS to use.

The use of PROC BUILD in the SAS/AF system and the use of PROC FSBROWSE (in the SAS/FSP system) when called from a SAS/AF system menu screen require more memory than is left on a 640K system logged on to the network. In practice, both the programming and the use of ALMA requires two megabytes of extended memory on the local workstation. This added memory must be managed by an extended memory driver to make the memory available to the SAS system. In addition, the EMS parameter in the CONFIG.SAS file must be set to ALL and the system (on the network server) must be installed with the option to use EMS.

These memory requirements limit the number of patrons who can use the system from their own workstations, although any patron could access the library from one of the "open area" PCs available at most locations. Given the high memory requirements we are currently reconsidering the strategy of providing patron access to PROC FSBROWSE through a SAS/AF system entry call.

The ALMA programs and collection databases are stored on network disks, chiefly so that they can be shared with other staff, and secondarily to take advantage of network file security and the nightly backup system. We have been successful in sharing the programs and the data files. However, we have also lost entire databases when multiple users with write access to the files have attempted to open a file for writing at the same time. The mechanics of this kind of event are still under investigation.

We have also determined that users with write access will lock users with read-only access out of a file. Because the library staff, who use ALMA frequently and for extended periods of time, would lock patrons out of the databases unacceptably often, we have set up a frequently refreshed database for the use of patrons, while "working" copies of the data files are kept in the library workgroup's file area.

For More Information:

Carmen Vincent or Linda Gowen
Westat, Inc.
1650 Research Blvd.
Rockville, MD 20850

SAS, SAS/AF, and SAS/FSP are registered trademarks of the SAS Institute, Inc., Cary, NC, USA.

Novell NetWare 386 is a registered trademark of Novell, Inc., Provo, Utah, USA.
Figure 1. ALMA Collection Selection Menu

Figure 2. ALMA Books and Manuals Subsystem Menu and Program Structure
Figure 3. Order Posting: ALMA Training Materials Example

+FSEDIT IN.NEWDP---Obs 1---

<table>
<thead>
<tr>
<th>Command</th>
<th>Status: 00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title:</td>
<td>ABSTRACT MACHINES &amp; GRAMMARS</td>
</tr>
<tr>
<td>Sub-Title:</td>
<td></td>
</tr>
<tr>
<td>Author:</td>
<td>Savitch</td>
</tr>
<tr>
<td>Co-author:</td>
<td></td>
</tr>
<tr>
<td>Shelving Code:</td>
<td>PR</td>
</tr>
<tr>
<td>Publisher Name:</td>
<td></td>
</tr>
<tr>
<td>Initials</td>
<td></td>
</tr>
<tr>
<td>Borrower's:</td>
<td></td>
</tr>
<tr>
<td>Requester1:</td>
<td></td>
</tr>
<tr>
<td>Requester2:</td>
<td></td>
</tr>
<tr>
<td>Requester3:</td>
<td></td>
</tr>
<tr>
<td>Check out date:</td>
<td></td>
</tr>
<tr>
<td>Order date:</td>
<td>08/25/90</td>
</tr>
<tr>
<td>Library Code:</td>
<td>01</td>
</tr>
<tr>
<td>Year Published:</td>
<td>1982</td>
</tr>
</tbody>
</table>

Description

Creates a new book/manual record and collects the author(s), title(s), publisher, publication date and requester's code. Order date and status are automatically filled. If the manuals are SAS publications (Shelving Code = SM), the publisher's name is filled in automatically.

Usage Highlights

<table>
<thead>
<tr>
<th>Function</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSEDIT</td>
<td>The FSEDIT SCL function executes a PROC FSEDIT procedure. The parameters are the data set name, the name of the FSEDIT customized screen, the mode of access (which may be EDIT, BROWSE, NEW, or ADD), and the OBS number of the first record to be displayed.</td>
</tr>
<tr>
<td>CUSOBS</td>
<td>CURSobs returns the current observation number. Its argument is the data set ID.</td>
</tr>
<tr>
<td>SYMGET</td>
<td>SYMGET is a macro function which returns a character value from the macro symbol table, similar in purpose to a global symbol table or global variables.</td>
</tr>
<tr>
<td>OPEN</td>
<td>The OPEN function opens a SAS data set. Its parameters are the data set name and the mode, which may be either INPUT or UPDATE. The Patron database is opened in the INIT section of the code below.</td>
</tr>
<tr>
<td>SET</td>
<td>SET associates variable names in the SCL program or on a screen with variable names in the data set.</td>
</tr>
</tbody>
</table>

```sas
init: call symput('order', '1'); return;
main: call fsedit('in.newdp', 'IN.BIBLIORARY.ORDER.EDIT', 'ADD'); return;
term: call display('ordrm.prg'); return;
fsedit: return;
init: orderdate = date(); odate = date(); status = '00'; callno = curobs(); return;
main: if key1 = ' then alarm;
if request1 = ' then do;
inistat = 2;
call symput('initchek', request1);
call display('in.dplibrary', 'chekinit.prg');
inistat = symfun('inistat');
if inistat = 2 then request1 = ';
end;
if request2 = ' then do;
inistat = 2;
call symput('initchek', request2);
call display('in.dplibrary', 'chekinit.prg');
inistat = symfun('inistat');
if inistat = 2 then request2 = ';
end;
return;
term: title = trim(title) || trim(title1) || trim(title2);
call execmd('end'); call execmd('end'); return;
fsedit: return;
init: if dsid = 0 then do;
msg = 'Initials file can not be opened - ALARM;
end;
end:
initstat = symfun('inistat');
if initstat = 2 then return;
return;
```