

# THE DEVELOPMENT OF A FLASHER TRACKING SYSTEM

Linda Ivey, Miles Inc., Agriculture Division

## ABSTRACT

A need was recognized for a system to plan, schedule and track research projects required for EPA registration of new products, to add new uses to existing products, and to comply with EPA data call-in requirements for existing products.

Within our Quality Improvement Process, a Corrective Action Team was formed to establish procedures for consistent handling and tracking of registration Flashers to reduce duplication of effort and to provide communication of testing and other action needed for regulatory items. There had to be a summary of EPA requirements, a current status of work being or to be performed, and the information had to be available on-line to potentially 25 users located at sites 50 miles apart.

Utilizing SAS/AF\*, SAS/FSP\* and SAS/SHARE\*, a system was developed that allows users the flexibility of editing a data set as well as creating or adding to the tracking documentation through a menu-driven facility. Host-connected HP II/III laserjet printers, installed with Avatar Passport-Plus boards are used for hard-copy output.

## BACKGROUND

A Flasher is a request from any regulatory agency for additional or new information or data relative to a registered product or to a product submitted for reregistration.

A regulatory Flasher goes through the following process:

- initiation
- summary
- analysis of action needed
- action assignment
- preliminary determination of status
- test/response proposed
- test/response approved
- test/response scheduled and initiated
- formal reply to Flasher
- Flasher follow up

These activities may be grouped by the person or Section responsible, as follows:

- REGISTRATIONS - initiation and summary
- TASK FORCE - analysis and action assignment
- ASSIGNED SECTION - preliminary status and test/response proposal
- TASK FORCE - test/response approval
- ASSIGNED SECTION - test/response schedule and initiation
- ASSIGNED SECTION - test/response status
- ASSIGNED SECTION - formal reply to Flasher
- REGISTRATIONS - reply to agency and Flasher follow-up

## REQUIREMENTS/CONSIDERATIONS

- decide who gets what when
- what sort of format
- must be a contiguous document
- must have search capabilities
- must have reporting capabilities
- equipment and training
- multi-user access/different site locations
- must be "user friendly"

## THE SYSTEM

The System was designed for use by some 25 novice users. SAS\* was chosen as the development tool because of the variety of its capabilities, including on-line entry and editing, search/select capabilities, and the flexibility of being menu-driven. Two primary concerns: one document in one place, updated by any of the users (FSLETTER), - the users being located at sites 50 miles apart (SHARE).

The System begins with a simple menu. By selecting '1' from the menu, FSEDIT is called to bring up the data file. At this point, the user can either modify an existing record, or create a new record.

If creating a record, certain data elements are automatically loaded for the user. We found this particularly useful for spelling consistency. Help screens as well as extended tables are provided where applicable.

The key to the data file is the Flasher Number, a unique 4 digit identification number assigned by our Registration Section.

Using Screen Control Language, PMenus are turned off to avoid user confusion. There are also several edits relative to required dates, etc.

When the user is ready to create the actual Flasher, an action individual is selected, together with the from individual, and an array is used to allow the user to choose up to 16 individuals for distribution copies. Credit for help with this particular array goes to SAS Institute Technical Support. See Exhibit I for array code.

The user chooses between creating a Flasher or modifying a Flasher by entering 'X' by elements called CREATE or MODIFY. See Exhibit II for code to create or modify a letter.

Within the FSLETTER Catalog there resides an "Initial" letter - a virtual blank form letter with the corporate header. When creating a Flasher, this initial letter is copied, using the Flasher Number as the key.

When modifying a Flasher, the existing Flasher is called for modification; again, the key is the Flasher Number.

The System Date is used for the Flasher date, and certain of the data elements are passed to the Flasher letter.

At this point, the user begins with the subject part of the Flasher, and writes whatever text is appropriate. After the Flasher has been sent to the printer, the user is returned to the data file, from which point he can either edit the data file, create another Flasher, or exit. Exiting the data file returns the user to the Main Menu.

A highlight of FSLETTER in our situation is

once the user issues the SEND command, and is instructed to make any final modifications, he could delete any verbage he wanted, sending only selected information, without actually eliminating verbage from the Flasher. (Changes made after the SEND command are not saved).

By selecting '2' from the Main Menu, FSEDIT is called to bring up the data file, using a different screen. The Part2 data file is for storing comments.

By selecting '3' from the Main Menu, FSEDIT is called to bring up a master file of product information. This master is used to load values into Part1 File. (See above).

By selecting 'RM' from the Main Menu, the user is sent to a Reports Menu from where he can choose a variety of reports. Generated reports come back to the screen; by pressing END, the user is given the opportunity to determine if a hardcopy is required. If so, a selection of printer locations is available. The user cursors to the desired printer designation, presses ENTER, and the hardcopy is sent.

The user is returned to the Main Menu after each selection.

A "go back" program is built into the System for easier development - transparent to the user. This simply means from the Main Menu, Proc Build is called to get to the actual code for developer modification.

#### SUMMARY

In our opinion, the products used in the development of this System are the best available on the market to date. The downside is in user training; most of our users had some familiarity with PC word processing tools which appear to be somewhat more flexible than the text processing of FSLETTER.

I would like to thank W. T. Pitts, A. D. Cohick, and J. L. Ivey for their help and support.

\*SAS, SAS/AF, SAS/FSP and SAS/SHARE are registered trademarks of SAS Institute, Inc.

HP LaserJet Series II and HP LaserJet III are trademarks of Hewlett-Packard Company.

Passport-Plus is a trademark of Avatar Corporation.

The author may be contacted at:

Miles Inc.  
8400 Hawthorn Road  
PO Box 4913  
Kansas City, MO 64120-0013

(816) 242-24902

## Array to Allow Up To 16 Distribution Copies of Letter

```
* ARRAY VALS{N} $1 NAM1 NAM2 NAM3 NAM4 ETC;
* ARRAY NAMES{N} $15 ('ACTUAL NAM1',
                     'ACTUAL NAM2',
                     'ACTUAL NAM3',
                     'ACTUAL NAM4',
                     'ACTUAL ETC ');
** ARRAY CC{N} $15 CC1-CC16;
    COUNT=0;
* DO I=1 TO N;
  IF VALS{I} = 'C' THEN COUNT + 1;
END;
** IF COUNT > N THEN DO;
** MSG = 'YOU CANNOT SELECT MORE THAN N FOR DISTRIBUTION";
  RETURN;
END;
** DO I = 1 TO N;
  CC{I}=' ';
END;
  J=0; ACTNUM=0;
* DO I=1 TO N;
  J+1;
** IF VALS{I} = 'C' AND ACTNUM <= N THEN DO;
  ACTNUM+1;
  CC{ACTNUM} = NAMES{J};
END;
END;
```

\* = WHERE N IS THE ACTUAL NUMBER OF CHOICES AVAILABLE

\*\* = WHERE N IS THE ALLOWABLE NUMBER OF CHOICES

### Exhibit I

#### FSP/SCL Code to create a Letter

```
IF MODIFIED(CREATE) THEN DO;
AAA = 'FLASHER.LETTERS.INITIAL.LETTER'
BBB = 'FLASHER.LETTERS.' || FLNO || '.LETTER';
RC=COPY{AAA,BBB,'CATALOG'};
IF (RC=0) THEN DO;
CALL EXECCMD('EDIT ' || FLNO || '.LETTER');
END;
END;
```

#### FSP/SCL Code to modify a Letter

```
IF MODIFIED(MODIFY) THEN DO;
CALL EXECCMD('EDIT ' || FLNO || '.LETTER');
END;
```

### Exhibit II



