The Development of an Integrated Tracking Application in an Information Systems Environment
Julianne Staub Perry, SAS Institute Inc., Cary, NC

ABSTRACT
Tracking problems that arise in an Information Systems environment is not a simple task but is very necessary to ensure a quality operation. Problems that should be tracked in Information Systems arise not only in the data center itself, but also in the areas of hardware communication and management information systems (which includes user services). All three areas need special attention, and though they differ in many ways, the basic tracking information is shared among them. Because of this, it is optimal to create a system that combines these areas into one Information Tracking System. Data are shared among each, yet they also contain information unique to their particular environment. It is also optimal to provide a way for incidents to be routed from one area to another without leaving the data set. Accomplishing these tasks involves the use of the powerful SAS® System. An example includes using SAS/FSP® to display the same SAS data set in three different ways. Another feature includes the use of formats and informats to aid in producing accurate information.

This paper briefly describes the importance of integrating the different areas of Information Systems into one tracking system application and summarizes the initial conversion process. However, its primary focus is on the current function, status, and special features of the application using SAS/AF®, SAS/SHARE® and SAS/SHARE® software, plus, the SAS macro language and a variety of SAS procedures.

BACKGROUND
The Information Systems Department at SAS Institute Inc. is divided into three areas of expertise. All three must focus attention on problems that users within the Institute encounter. Originally, this was accomplished by developing three different menu-driven applications using SAS/AF software to track the problems called in by users, or in the data center case, problems involving various operating systems.

Here are the three areas of expertise and the corresponding tracking system that was used to track their problems:

Communication Services - COMTRAK
Management Information Systems and User Services - MISTRAK
Data Center Operations - DCTRAK

Communication Services' primary function is to maintain, facilitate, and install the various computer and telephone systems throughout the company. The tracking system's purpose is to track and report any calls made to them either through the help desk or user services or calls made directly to the Communication Services department. The tracking application informs technicians when a problem occurs in their area.

The Management Information Systems and User Services department's purpose is to create, maintain, and help users with SAS applications, operating systems, and general hardware and software questions and problems. The tracking system's ease-of-use and durability are important, its primary functions are to log any calls made to the department for historical purposes and to act as a tool to indicate problems that must be taken care of immediately. Because calls to the help desk are often transferred to another area of expertise, it is essential that the data entered from the initial call route Information to the other areas quickly and easily. The ISTRAK concept allows this.

The Data Center Operations department monitors and controls all aspects of system activity in the data center. The goal of the tracking system is to record any abnormal activity that occurs and to use that information as a means of correcting a problem, finding any trends in system malfunctions, and scheduling system downtime.

INTRODUCING ISTRAK
Within the past seven years, Information Systems has grown tremendously. As Information Systems grows, so does the amount of information that needs to be tracked. The managers had a greater need for monitoring system and user problems. This was a difficult task because each of the applications varied greatly (primarily due to the different variable values in each SAS data set). It was also frustrating when a problem common to all of the areas of expertise needed to be tracked. Through analysis, a solution became apparent: The applications could be converted and merged into a single master application revolving around one master SAS data set. This would particularly lend itself to situations in which problems required expertise from more than one Information Systems group. It would allow flexibility in referring problems to alternate sources. In other words, if all of the information were stored in a single SAS data set, then the data could be analyzed or viewed more efficiently and effectively. Thus, the concept of ISTRAK was born.

Why the SAS System was the Perfect Answer
The SAS System was chosen as the primary language for a variety of reasons. First, and probably obviously, user familiarity with the language's end-user concepts is high and also preferred within the Institute. The user is not required to know the SAS software language because of its front-end procedures. Second, it is the historical precedent. Third, SAS software offers compatibility with other applications used throughout the MIS department. Thus, necessary interaction between data sets is made simple. Fourth, SAS is extremely versatile and its products fulfill most every need we could foresee in our tracking system's goals and objectives. And finally, SAS offers flexibility. The SAS System can handle any changes or alterations that are made to accommodate each area's changing needs.

Merging the Three Tracking Systems
A plan was developed to merge the three primary Information Systems tracking applications, COMTRAK, MISTRAK, and DCTRAK, that produced the final and new application, ISTRAK. The following is a brief summary indicating the steps taken to merge the three existing data sets that formed ISTRAK:

1. Review the contents of each master data set, identifying duplicates (common variables).
2. Design conversion of the data sets.
3. Convert the data sets.
4. Design the screens.
5. Develop prototype systems for user review and input.
6. Create and run the conversion jobs.
7. Revise the SAS programs.
8. Test the subsystem applications before putting them into production.
9. Put the subsystem applications into production.

ISTRAK SYSTEM REQUIREMENTS

System requirements were defined before the merge took place. Four major requirements were discussed and agreed upon.

- **Variable Continuity**: Common variables are shared among the three operational areas: COMTRAK, MISTRAK, and DCTRAK. For example, each area is concerned with who reported the incident, who the current owner of the incident is, when the incident occurred, a brief description of the incident, when it was closed, and so on. Each of the variables are visible and labeled consistently on each screen. Yet, a variety of other variables unique to the area should also be visibly customized on the screen.

- **User Friendly**: The user should not be required to know the SAS language. However, use of SAS/FSP searching techniques is required. They are simple enough to master. SAS/AF is used as a front end and offers easy-to-use report menus.

- **Flexible**: The data set and the reports must be easy to alter and customize according to the needs of each area of expertise.

- **Multi-User Access**: SAS/SHARE software should be used to allow for multi-user access. This enables users and experts to communicate interactively.

ISTRAK SYSTEM DESIGN

ISTRAK runs under MVS with a menuing system using SAS/AF software as its user interface. Data entry, reporting options, and error checking are provided on this level. Each Information System's area owns and invokes its own primary menu. This allows the users in each area to concentrate solely on their function, yet also allows users interested in the application as a whole to have access to the other elements of the system. Figure 1 (see the end of the paper) is a flow diagram overview of the design of ISTRAK and how each operational area interacts with another.

**System Structure**

The organization of the master data set shared among each area of expertise is straightforward. In our application, it was chosen to sort the data by the area of expertise and then by a preassigned incident number. This is simply because during a typical day of data entry, end users remain in their own block of observations. Note that the "daily observations" are predefined and are simply "blank" ready for information to be added to them. Refer to Figure 2 at the end of this paper. The values that are defined when the "blank" observations are defined are done to prevent the user from having to enter repetitive data and to ensure correct values when they are already known. For example, the "Date Entered" field is filled in nightly so the user does not have to worry about entering the date each time he/she receives a call. Considering the user services group receives hundreds of calls per day, this saves a lot of time. The observation number values are defined nightly and are displayed to the user so that the user may jump around the data set quickly.

When initial entry to the master data set is invoked, the user is automatically placed in his or her area of expertise. Appendix A shows a block of code used to determine where the user should be placed at initial entry.

**Data Entry**

The master data set uses the SAS/SHARE file server so that multiple users may enter data simultaneously and remain in the SAS data set throughout the working day. Also, each area's screen is designed to fit its purpose of fast and efficient data entry. During a session of entering data, it is important to set up the SAS/FSP screen according to the needs of each area to ensure accurate data that can be entered quickly. But also important is readability for those who simply need to browse the data.

**Reporting Options**

The ISTRAK master menu branches off in three directions. Each direction represents the three areas of expertise as described above. In each sub-menu, reporting options are available for the user. Appendix B, Screens 1 through 3 display the types of reporting options the user may choose. Printing options are offered directly from each panel or menu. Options to print reports are also offered from sub-menus of each main menu. Appendix B, Screen 4 displays the reporting sub-menu branched from the Data Center sub-system. Appendix B, Screen 5 is an example of the panels displayed on a reporting option and is selected directly from one of the main sub-systems.

**Error Checking**

Nightly jobs are run to check any data entry errors that may have been made during the day. Examples of checks are discussed below.

Because incident numbers are automatically defined nightly, duplicate ones are not often created accidentally. However, duplicate numbers have been detected and thus, must be taken care of. It is important to make sure that any duplicate numbers created are assigned new numbers as soon as possible to prevent confusion to both the system user and the person reporting a problem.

Dates are checked for validity. For example, a user may indicate an incident has occurred on 01APR90 at 15:00:00 and claim it was over on 01APR90 at 07:00:00, intending for the 7:00 to mean p.m. The job knows that 15:00>07:00 so the user must mean 19:00 and flags the incident as having a data entry error.

Other data entry inconsistencies are also checked. Version 5 of SAS/FSP software does not allow interactive data checks and some fields need to be filled in if others are. The error checking jobs ensure this or at the very least make sure that someone is notified when inconsistencies occur. Note, however, Version 6 will allow immediate error checking upon entry of a value.

SAS informats are also used as a means of error checking. They check that the value entered is appropriate for the variable. These are discussed in more detail in the next section.
ISTRAK SYSTEM FEATURES

In order to create a functional, easy-to-use, and accurate data base, it was important to design features to aid the user entering data. The following are a few examples of features used to accomplish this task.

Customized Function Keys

During invocation of ISTRAK, customized keys are copied into a user profile (SASUSER). The ADD function key is replaced by a customized Function Keys an observation when necessary. Appendix C describes how the user profile is created and invoked.

Initial Entry According to Tracking Application Chosen

Appendix A shows an example program that decides what observation the user should land on when entering the ISTRAK master data set. This feature sets bounds for the user to work in during the day. Yet, the user is free to move around the entire data set if desired.

Use of SAS Formats and Informats

Many of the variables in the master data set contain both formatted and informated values. For example, the status variable only allows an O for open and a C for closed.

Examples of informated values include the 'caller' and 'callfor' fields. A job runs nightly to determine Institute employees in-house extension and places the value of their extension, name, and room number when the extension is entered in the field. This aids in easy data entry and quick reference to the caller. Other examples of formatted values are the fields 'Failures Testing Component', 'hardware and software', and 'Servicell Unavailable'. Again, a nightly job runs to pull information from a variety of partitioned, sequential, and SAS data sets to comprise a set of valid values. A code is produced so that the user simply enters the code, and the more detailed information is displayed. Appendix E supplies some code that creates an informat by reading from a SAS data set.

Searching Techniques

All SAS/FSP searching techniques apply in this application. The search modification 'Field Attribute' section is defined to automatically string on the variables each area of expertise uses most often. This is another advantage of using three different screens to represent the same data.

Incident Owner Notification

The owners of incidents are notified about incident existence through several techniques. The first is that a nightly job runs to produce listings indicating which incidents are open for a particular owner and a brief summary about each incident. Summary jobs use different techniques depending upon the sub-menu the user selects. Examples of SAS procedures include: PROC PRINT, PROC TABULATE, PROC GPRINT, PROC FREQ, and PROC SUMMARY.

Our electronic mailing system is a second technique used to notify owners of their open incidents. Hourly and nightly batch jobs send summary messages to the owners.

And finally, the reporting options on each menu offer a way to run listings or browse data according to the way the user likes to view it.

Using PROC FSLETTER to Print Observations Immediately

In our ISTRAK application, the Data Center needs to send single observations directly to a printer for away from the terminal reference. Thus, it was necessary to set up an FSLETTER catalog to allow it to do so. The catalog contains a letter that looks very similar to the data center screen. Appendix A indicates a way to allocate an FSLETTER catalog depending on the area of expertise chosen. Users may then send an observation directly to the printer if desired. Appendix F is an example of an FSLETTER letter used to produce the printout of an incident for the Data Center Operation's staff.

CONCLUSION

The ISTRAK application's primary function is to track problems in Information Systems, whether the problems fall in the operational areas of management information systems and user services, communication services, or data center operation, or all three. This is possible through the use of SAS/SHARE, SAS/FSP and SAS/DIF software. If anything is gained from this paper, it should be the importance of having a single data set represent the three areas of expertise in what appears to be three different ways. The single data set allows for easy access and data manipulation. It is versatile in that the data may be treated as a single entity or may be easily divided into categories. It is flexible in that changes can be made to the data quite conveniently.

The ISTRAK System Application runs under a TSO environment. This is not to say, however, that its function could not be implemented in an alternate environment. The concept of one data set used for three different applications is the same, regardless.

APPENDIX A - Sample Program

The code provided below is an excerpt from the ISTRAK application. It is used to determine the observation number the user should land on when entering the master data set. The number is assigned depending upon the sub-menu the user selects.

&MACRO chkobs;
	DATA ...NULL...;
	RUN;
&END;

&MACRO chkobs;
	DATA ...NULL...;
	RUN;
&END;

&MACRO chkobs;
	DATA ...NULL...;
	RUN;
&END;

&MACRO chkobs;
	DATA ...NULL...;
	RUN;
&END;

&MACRO chkobs;
	DATA ...NULL...;

Using PROC FSLETTER to Print Observations Immediately

In our ISTRAK application, the Data Center needs to send single observations directly to a printer for away from the terminal refer-
ence. Thus, it was necessary to set up an FSLETTER catalog to allow it to do so. The catalog contains a letter that looks very similar
to the data center screen. Appendix A indicates a way to allocate an FSLETTER catalog depending on the area of expertise chosen. Users may then send an observation directly to the printer if desired. Appendix F is an example of an FSLETTER letter used to produce the printout of an incident for the Data Center Operation's staff.

CONCLUSION

The ISTRAK application's primary function is to track problems in Information Systems, whether the problems fall in the operational areas of management information systems and user services, communication services, or data center operation, or all three. This is possible through the use of SAS/SHARE, SAS/FSP and SAS/DIF software. If anything is gained from this paper, it should be the importance of having a single data set represent the three areas of expertise in what appears to be three different ways. The single data set allows for easy access and data manipulation. It is versatile in that the data may be treated as a single entity or may be easily divided into categories. It is flexible in that changes can be made to the data quite conveniently.

The ISTRAK System Application runs under a TSO environment. This is not to say, however, that its function could not be implemented in an alternate environment. The concept of one data set used for three different applications is the same, regardless.

APPENDIX A - Sample Program

The code provided below is an excerpt from the ISTRAK application. It is used to determine the observation number the user should land on when entering the master data set. The number is assigned depending upon the sub-menu the user selects.

%MACRO chkobs;
	DATA ...NULL...;
	RUN;
&END;

%MACRO chkobs;
	DATA ...NULL...;
	RUN;
&END;

%MACRO chkobs;
	DATA ...NULL...;
	RUN;
&END;

%MACRO chkobs;
	DATA ...NULL...;

Using PROC FSLETTER to Print Observations Immediately

In our ISTRAK application, the Data Center needs to send single observations directly to a printer for away from the terminal refer-
ence. Thus, it was necessary to set up an FSLETTER catalog to allow it to do so. The catalog contains a letter that looks very similar
to the data center screen. Appendix A indicates a way to allocate an FSLETTER catalog depending on the area of expertise chosen. Users may then send an observation directly to the printer if desired. Appendix F is an example of an FSLETTER letter used to produce the printout of an incident for the Data Center Operation's staff.

CONCLUSION

The ISTRAK application's primary function is to track problems in Information Systems, whether the problems fall in the operational areas of management information systems and user services, communication services, or data center operation, or all three. This is possible through the use of SAS/SHARE, SAS/FSP and SAS/DIF software. If anything is gained from this paper, it should be the importance of having a single data set represent the three areas of expertise in what appears to be three different ways. The single data set allows for easy access and data manipulation. It is versatile in that the data may be treated as a single entity or may be easily divided into categories. It is flexible in that changes can be made to the data quite conveniently.

The ISTRAK System Application runs under a TSO environment. This is not to say, however, that its function could not be implemented in an alternate environment. The concept of one data set used for three different applications is the same, regardless.

APPENDIX A - Sample Program

The code provided below is an excerpt from the ISTRAK application. It is used to determine the observation number the user should land on when entering the master data set. The number is assigned depending upon the sub-menu the user selects.

%MACRO chkobs;
	DATA ...NULL...;
	RUN;
&END;

%MACRO chkobs;
	DATA ...NULL...;
	RUN;
&END;

%MACRO chkobs;
	DATA ...NULL...;
	RUN;
&END;

%MACRO chkobs;
	DATA ...NULL...;
LIBNAME is CLEAR;

"TSO FREE FILE(screen isndx); RUN;

%IF isname=OC THEN %00;
UN;
UND;
%MEND isedit;
APPENDIX B - Sub-menus of ISTRAK

Screen 1 Management Information Systems and User Services Sub-menu

Screen 2 Communication Services Sub-menu

Screen 3 Data Center Sub-menu

Screen 4 Reporting Menu from the Data Center Sub-menu
Screen 5 Sample Reporting Panel for Print Options

APPENDIX C - Creating a User Profile

This is a step-by-step process indicating how to set up user defined function keys.

1. Create a permanent SAS data set.
2. Allocate the user profile with ddname=SASUSER.
3. Edit via PROC FSCALC.
4. Edit via PROC FSCALC CAT=SASUSER.PROFILE; RUN;
5. Copy SASUTL.FSP.FSEDIT.KEYS into the user-defined SAS data set.
6. Select FSEDIT.KEYS.
7. You may change any of the function keys and use things like the FIND command to search for variable values.
8. Exit the profile.
9. The FSEDIT keys should now be set.

Every time you wish to use these settings, the function key SAS data set must be accessed with ddname=SASUSER to replace the temporary profile before entering the SAS System.

If you want users of a specific system to allocate the function key permanent profile into a work one. This is because in order to use the permanent profile, SASUSER must be accessed each time they execute the GLST. You must copy the permanent data set and a SAS macro used to copy the profile over.

Below is an example of the CLIST allocation of a temporary SAS data set:

ALLOC FSP(SASUSER) SP(1) TRACK MYPAGE
ALLOC FSP(SEVEN) DA('PERMANENT,SASUSER') SAS USER
SET IS=ISPLS,X=ISPLSPROF, SCISPRAK, RMANAS, RUN(+)
SAS OPTIONS('V13 MODUS')
AUTOS('ALL,ISPLS,NMS','')
FREE FSP(SASUSER)

MACRO used to copy profile:

MACRO bidprofile;
PROC CDS FN=p(euser) OUT=userasuser;
SELECT profile;
RUN;
STOP FREE FSP(p(euser));
ENDM bidprofile;

APPENDIX D - Screens Used to Represent Each Area of Expertise

Screen 6 Management Information Systems and User Services Screen

Screen 7 Communication Services Screen

Screen 8 Data Center Screen 1
Screen 9 Data Center Screen 2

APPENDIX E

Code used to produce informats by reading a SAS data set.

```sas
//INPUTFORM JOB ('xxxxx','.'/S.2021'),TIME=(',S'],'MSGCLASS=I 00010020
//*SASLIB FETCH 00020026
// EXEC DAS 00030020
//SASLIB DD DSN=SAS.ALOYEE.DATA,DISP=SHR 00040034
//ENWRF DD DSN=SAS.ALOYEE.DATA,DISP=SHR 00050040
//CCODIS DD DSN=SAS.ALOYEE.DATA,DISP=SHR 00060040
//SCIN DD DSN=SAS.ALOYEE.DATA,DISP=SHR 00070040
PROC SORT DATA=empdb.data OUT=employee; BY extension;
RUN;
DATA _NULL_
SET employee END=EOF;
FILE codes1;
IF _N_=1 THEN
PUT 'PROC FORMAT DASLIB='@; DEFAULT $CALLFMT';
PUT '1 extension ' @';
IF eof THEN GO;
PUT 'A3' ' ' ' ' ' _SAME_'@;
PUT 'OTHER ' ' ' ' _ERROR_'@;
END;
INCl codes1/NSOURCE2;
RUN;

APPENDIX F

The following is an FSLETTER screen representing the FSP screen which is used to send information to a printer. A brief description is also supplied indicating how to send the letter.

To send a letter, find the incident you want to print. On the command line enter SEND FORM2. The printed form will appear on the screen. If you decide not to send the form you can enter CANCEL on the command line and the original SAS/FSP screen will appear. If you decide to complete the command and send the form to the printer, you simply press PF15 (exit command) twice. If you want the form to print immediately, the following commands should be executed beginning from the SAS/FSP screen.

* CAT - Displays the letter catalog
* FREE - Frees the form
* PF15 (exit) - Returns to incident
Figure 1 Overview of the ISTRAK design
<table>
<thead>
<tr>
<th>Data Center Operations</th>
<th>Communication Services</th>
<th>MIS/Help Desk</th>
<th>Daily Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>doobs = Number of observations in the area of Data Center Operations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>comobs = Number of observations in the area of Communication Services</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>misobs = Number of observations in the area of MIS/Help</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2  Organization of Master Data Set