Managing SAS/AF® Application Development on a Local Area Network

Michael Kilhullen, Sandoz Pharmaceuticals Corporation, East Hanover, NJ
Andrea Contino, Contino Associates Inc., North Caldwell, NJ

ABSTRACT

As a Lead Application Developer, it becomes important to know the development status of each application without having to spend time to meet with each team. There is a need to ensure that work is not being duplicated, resources are properly allocated, change control is maintained, and standards for GUI interface design are satisfied. This paper describes a SAS/AF® Frame Entry application designed to satisfy these requirements, expand upon automation of development, unit testing, implementation, and version control. In addition, a number of tools are provided in the application to generate status and summary reports on the development progress and production maintenance. The application takes full advantage of local and centralized development, testing, and production areas on a Local Area Network. Currently, the application focuses on the management of SAS® catalogs, however, it is envisioned that the application will be expanded to manage SAS databases, macros and format libraries.

THE NETWORK ENVIRONMENT

Successful application development depends on a manageable directory structure. In the network environment, application development, testing, and maintenance occur on a shared network drive. To achieve this, the drive is divided into three primary directories (see table 1). The DEVELOP directory contains all source code for the development phase of the application. The PRODUCT directory contains the production version of the application. The TEST directory contains a production ready version of the application that requires user testing. Only the DEVELOP directory has write access. Any changes to an application must be made in DEVELOP, then copied to the other directories by the Application Administrator. Finally, access and privileges to these directories are determined by network user groups.

Table 1 - Application Development Primary Directories

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVELOP</td>
<td>Contains all development code for applications</td>
<td>Application Developers read/write</td>
</tr>
<tr>
<td>TEST</td>
<td>Contains a production ready version of the application for testing</td>
<td>Selected Users read only</td>
</tr>
<tr>
<td>PRODUCT</td>
<td>Contains the production version of an application</td>
<td>All Users read only</td>
</tr>
</tbody>
</table>

Standardizing SAS Application Directories

To automate management of SAS/AF Applications, a standard directory structure must be implemented to store all of the application components. In addition, this directory structure should be identical between all three primary directories (see figure 1). This makes the movement of applications through the development process quite easy. There are many ways you can standardize your directory structure, however, the application described in this paper depends on the following:

`/main application directory/CATLIB - catalogs`
`/main application directory/MACLIB - macros`
`/main application directory/DATALIB - static data`
Obviously, these directories do not cover all SAS components, however, the basic directory structure allows for easy integration of new directories.

Figure 1 - Sample Application Installed In The Network Directory Tree

The APPMAN Application

Managing a SAS Application occurs in the DEVELOP area. This is the focus of APPMAN, a SAS/AF Application designed to control the development of SAS applications in the network environment. APPMAN controls everything from setting up development application directories to moving a final application into production.

The Directory Management Process

Within APPMAN, the Lead Application Developer indicates which directories require management. By default, all standard directories are automatically setup when a new application is added. For each of these directories, APPMAN adds MASTER and INTERMED sub-directories. The MASTER directory contains all final versions of the directory being managed. The INTERMED directory contains all development programs contributed by Developers that require approval for addition to the MASTER directory. A continuous flow occurs between these two directories until an application is ready for production (see figure 2).

Role of the Lead Application Developer

The Lead Application Developer is responsible for controlling the flow of programs from the INTERMED area to MASTER and for moving an application between the DEVELOP, TEST, and PRODUCT areas. When a Developer adds a program to INTERMED, the

Lead Application Developer is notified within APPMAN. The programs added are automatically copied to the Lead Application Developer's private test area where he can review the code, frame layout, etc. If the program is satisfactory, it is moved to the MASTER directory. However, if a problem is found with the program or its documentation, it is returned to the contributing Developer. At this time, the Lead Application Developer is allowed to comment on the reasons for the programs rejection, and provide recommendations for correcting the problems.

Role of Application Developers

Application Developers are responsible for developing the frames, menus, macros, etc., that make up the application. The bulk of the work is done in a development area located on the Developer's private network drive. For programs that already exist in MASTER, the Developer can 'check out' selected programs. Once a program is checked out, no other Developer can select it, thus, ensuring that work is not being duplicated. The program is copied from MASTER to the Developer's private drive where changes and enhancements can be made. At the same time, new programs can be added. Once programs are completed, they are interactively returned to the INTERMED directory and the Lead Application Developer is notified.

Using APPMAN

When APPMAN starts, the main project selection box is displayed (see figure 3). The available projects will appear in the list box. Buttons allow the Developer to use development tools (copy, check-in, check-out, etc.), generate reports, add/modify projects, and exit to development and test environments. The remainder of this paper will discuss the key modules of the application.
Development Tools

The Development Tools Window provides the key functionality of the application (see figure 4). From this menu, Developers can perform the necessary steps to build the application. The Lead Application Developer is provided selections to effectively manage the development process.

Checking Out Programs

The Application Developer is provided a list a catalogs and programs that are available in the MASTER directory (see figure 5). By checking out a program, the Developer is the only person who can make changes to the program. This is primarily to avoid duplication of work. The selected programs are automatically copied to the Developer's private work area.

Returning Programs to MASTER

Once the Developer has completed modifications to a program or has created new programs to add to the application, the next step is to "check in" the programs (see figure 6). A list of all programs checked out by the Developer, including new programs, is displayed. The Developer can either check-in the program or abort changes and start all over. By aborting changes, the original version of the program will immediately be made available to other Developers. Otherwise, the Lead Application Developer will be notified that a program requires validation.

Validating Changes and Additions

Once a program is checked in, the Lead Application Developer must verify that the new programs meet company standards. From the Development Tools Window (see figure 4), the Lead Application Developer can select "Validate New Programs". Selecting this option will copy
all of the programs in the INTERMED directory to the Lead Application Developer's test area. He can then proceed to test the programs, view the source code, etc. When testing is complete, the programs are either approved or returned to the Developer for further revisions (see figure 7). In addition, the Lead Application Developer can also comment on the rejected programs to help the Application Developer with improvements. If approved, the program is copied to the MASTER directory.

![Figure 7 - Outcome of Validation Process](image)

**Returning Unacceptable Programs**

As mentioned previously, the Lead Application Developer has the opportunity to evaluate each Developer's programs during the validate process. Programs that do not meet up to application standards can be returned to the Developer by moving them to the "Entries to be Returned" box (see figure 7). When this is done, a new window appears which allows the Lead Application Developer to optionally comment on the problems found and suggest possible solutions (see figure 8). The next time a Developer uses APPMAN, he will be notified that programs are being returned.

![Figure 8 - Returning Programs](image)

**Completing an Application**

Once the application is complete, the Lead Application Developer can choose to move it into the TEST or PRODUCT environments. Once again, this is an automated process complemented by databases which track the application's version and revision history.

**Databases Controlling the Application**

A number of SAS databases are responsible for managing the available applications and tracking changes to the programs, versions, etc. The following summarizes these databases:

- APP - stores the development applications that are available in the Application Manager.
- APP_DIR - stores the libnames, filenames, and paths for each of the development applications.
- REPORTS - stores report definitions.
- STATUS - stores status information on all programs in development and/or production for a given application.
- UPLOADED - stores all programs that have been approved by the Lead Application Developer and exist in MASTER.
- CHANGES - audit trail that stores changes made to programs that are modified from MASTER.
- VER_INFO - tracks the version numbers assigned to an application.
- VERSIONS - tracks the individual programs that make up a version of an application.

**Other Application Features**

APPMAN also provides a multitude of other features for report generation, administration, and version control. The following section summarizes some of the more relevant features.

**Developer Notification**

In the project selection window, an asterisk will appear in the list box next to the application name to indicate to a Developer that their involvement is required. Since this list box is the first window seen in the application, Developers will immediately sense where their assistance is needed across all applications.
Exit to Development or Testing Areas

All application specific libnames and filenames are assigned when exiting to the users personal development or testing area. This eliminates the need for separate icons for each application and significantly streamlines the development process. Once selections are made, the development environment is only a click away.

Overwrite Protection

Because APPMAN is controlling the movement of files, overwriting or loss of files is eliminated. When a Developer copies the MASTER directory to refresh the private work area, files that are checked out are not overridden. Likewise, the MASTER version of a file is never replaced until changes are approved by the Lead Application Developer. This allows a Developer to "throw away" changes made to a program and start from scratch. The check in/check out process also ensures that Developers cannot overwrite each other’s work.

Version Control

After the first release of an application, changes to the programs are monitored. When a program is added to MASTER, the Lead Application Developer provides information regarding the changes made and reason for change. This is complemented by the ability to cross-reference this information with other applications or forms which track problems with applications. In addition, each time the production version of the application is updated, a record of all files that make up the version is kept. Because of these features, detailed audit reports can be generated.

Report Generation

A variety of log, audit, and summary reports have been built into the application. The report generation module was designed to allow new reports to be easily added. Reports can be displayed on the screen or sent to a network printer.

Reports Include:

- All entries found in the Master Development Catalogs
- Status Report: Entries Checked out/Checked in
- Audit Report: Entries Uploaded to the Master Development Catalogs
- Production Version Report
- User Test Version Report
- Contents of the Production Version

Application Administration

- Setup New Applications - The Lead Application Developer supplies a description, directory name, and list of Developers
- Automatic Directory creation - each time a component is added to the application, directories are automatically created on the shared and local drives through all the development environments.
- Create new catalogs - catalog files are automatically added in the DEVELOP, TEST, and PRODUCT environments.
- Define Developers - adding or changing Developers
- Automatic Libnames/Filenames assignments - defines which libnames and filenames to allocate when a Developer exits to development or testing areas from main menu.
- Update Application Versions - assign version numbers to test and production applications.
- Version Control - Document all changes made to a production version.

CONCLUSION

APPMAN demonstrates the benefits of automating the SAS/AF application development process. It provides the tools necessary to effectively manage the development process and monitor compliance with application development standards. Work is not duplicated, resources are properly allocated, change control is enforced, and a common interactive environment for all applications is maintained.

For more detailed information regarding this application or setting up a standard network directory structure, contact

Michael Kilhullen
Sandoz Pharmaceuticals Corporation
59 Route 10
East Hanover, NJ 07936-1080
Voice: (201) 503-7574
Fax: (201) 503-6363
email: kilhullen.michael@ehcglobal.sandoz.com

ACKNOWLEDGMENTS

I wish to acknowledge Dominick Scapati and Philip Rutledge for establishing the standard application development environment and standard programming guidelines. I would also like to thank Harvey Monder for his assistance in preparing this paper.

SAS and SAS/AF are registered trademarks or trademarks of SAS Institute, Inc. in the USA and other countries.
® Indicates USA registration.