LOAD DATA ANALYSIS WORKSTATION (LDAW): A SAS/AF® INFORMATION SYSTEMS APPLICATION

Burtt Blodgett, Quantum Consulting Inc.
John Powers, Quantum Consulting Inc.
Diane Bailey, Quantum Consulting Inc.

The Load Data Analysis Workstation (LDAW) is a user-friendly menu-driven information system that utilizes a wide variety of SAS Institute software products. Designed and developed by Quantum Consulting for the Electric Power Research Institute (EPRI), the LDAW enables electric utility staff to quickly analyze energy usage, load, weather and survey data from residential, commercial and industrial customers. The LDAW is available through EPRI to its member utilities.

In this paper we discuss a generalized version of the LDAW, emphasizing the innovative uses of SAS® software products rather than the detailed technical aspects of electric utility load data analysis. The remainder of this paper is organized as follows. First, we discuss our reasons for developing the LDAW and how it compares to existing mainframe-based systems. Second, we review the LDAW system design in terms of hardware, system software, and the array of SAS products used. Next, we describe the LDAW's user interface in some detail; this interface is unlike any of which we are aware written in SAS/AF®, and contains several innovations which may be of value to others developing SAS-based information systems. We then describe the capabilities of the analysis code, which is contained entirely in external SAS files so that users can modify these programs without recompiling the interface. Next, we describe a few of the "tricks" we used to pull these diverse elements together into a single system, including an undocumented feature of SAS/AF, known as compiled SCL macros. We conclude with a discussion of our experience in porting this application to other operating systems, and some general comments about the use of SAS/AF for developing information systems.

A Few Words About Load Research

Electric utilities have an unusual market research problem; although they know exactly how much of their product each customer uses, they have very little information about how their product is used. In order to gather information about the time of use and reasons for use of electricity, utilities can gather very detailed information from a small sample of their customers, using special meters. These meters gather electricity usage information once every 5 or 15 minutes (versus once a month for billing meters). The data gathered by these meters are referred to as "load data." Some of these meters can also be installed inside the customer's building to meter specific appliances or equipment, so that the utility will know the usage patterns for each "end-use" of its product. The analysis of data from these meters is known in the utility industry as "load research."

Some form of load research is undertaken at nearly every large electric utility and at many smaller utilities as well. The datasets involved in an average load research application are fairly large: typically in the tens of megabytes for each month of analysis. Additionally, such applications may require combining load data with auxiliary data such as customer survey information, weather data, and billing information. This research is almost always undertaken on a mainframe computer using programs written by each utility's own staff. At most utilities, the systems currently in use rely at least in part on SAS programs. The collection of SAS programs for load research around the country represents a minimum of several hundred person-years of software development time, much of it redundant. Each time a new programmer enters a load research department, he or she must learn the quirks and conventions of all the programmers that have worked on that utility's system to date before adding new capabilities.

Our idea in developing the LDAW was to bring load research capabilities to a workstation environment, while designing the system flexibly enough so that we could tap into the existing base of mainframe programs with a minimum of difficulty. Our goal was to develop a set of core capabilities, emphasizing flexible data input and output with some basic analytic functions, and to put them into a user interface that would dramatically shorten the "learning curve" of new load researchers. The response we have received to date indicates that we have succeeded; the LDAW is beginning to win acceptance in the utility industry, and plans are underway to establish several means for LDAW users to share analysis and reporting programs.

LDAW System Design

The LDAW is designed to operate on a high performance microcomputer system with a large data storage capacity and graphics capability. Figure 1 provides an overview of a typical LDAW system configuration. Data can be exchanged through a tape drive, over a local area network (LAN), or through a direct connection with a mainframe. An internal modem provides additional data support. LDAW analysis results can be output to a high resolution color monitor, as well as to any printer, plotter, or other output device supported by the SAS system.

System Configuration

The LDAW software serves as a user-friendly front-end to SAS analysis programs, as shown in Figure 2. User input parameters required for the programs are entered through the LDAW program input screens developed using SAS/AF® software. Screen Control Language (SCL) code checks all input for errors and assists the user through pop-up selection lists, error messages, and help screens. Correct input parameters are passed as macro variables to the external SAS files containing the analysis code. These external files make use of base SAS, SAS/STAT®, SAS/GRAPH®, and other SAS software products. Analysis results can be stored as SAS datasets, output as graphics, or transformed into a variety of popular formats.
How the LDAW Works

**SAS/AF® User Interface**
- User input
- Selection lists
- Error messages
- User help

**SCL Code**
- User input validation
- Macro variables representing user input

**SAS® Analysis Programs**
- Access data
- Validate data
- Analyze data
- Edit/modify data
- Create graphics

**Analysis Results**
- Output: SAS datasets, spreadsheet files, graphics

_Figure 2_
The LDAW has been designed in an open format for easy modification. Any of the SAS analysis programs can be changed without recompiling the user interface SCL code. Modifications to the user interface can be made readily and incorporated into the menu system with a minimal amount of effort. LDAW's open design gives it considerable flexibility; it can be adapted easily to a wide array of interface and analysis needs.

User Interface

The LDAW features a user interface that guides the user through analysis tasks by means of a menu system designed to be familiar to users of other pull-down menu systems. This menu system is unlike either the "icon" menu found in SAS/ASSIST® or the 1970's style "number choice" menu scheme supported by SAS/AF Version 6.03. Instead, the options of the main menu are arranged along a thin bar at the top of the screen. Selections are made by first tabbing to the desired option and then pressing <ENTER>. A mouse can also be used to make selections by clicking on the desired option. A submenu listing available options then appears in the form of a rectangular box below the main menu bar. The main menu bar remains on the screen. The user makes selections in much the same manner as the main menu; the user first tabs to the desired option, but now presses <ENTER> once to highlight the option, then again to select that option. The visual presentation is similar to that developed at Xerox PARC and made popular by Apple Macintosh® applications.

The order of the options along the main menu bar parallels that of the logical steps required to progress from raw input data to validated analysis results. System defaults are set, data is read into a SAS dataset from a raw input file, and then cross-checked for validation purposes. Next, analyses are performed, and the final results are either output as reports, graphs, datasets, or as export files. Submenus for each group of tasks appear beneath the corresponding option on the bar. Figure 3 shows a submenu as it appears in the user interface. As options further along the bar are selected, their corresponding submenus appear further across the screen. This visual ordering aids the user in remembering the location of specific programs within the system, and reinforces the logical progression of tasks.

Options on the submenus permit access to screens accepting input for the analysis programs. A typical input screen is displayed in Figure 4. On a color monitor the screens appear in a color-coded scheme to aid the user. The user is reminded which submenu the current program is under by a cyan bar at the top of the screen. Yellow text directly beneath this bar gives the full name of the analysis program. All fields that accept user input appear in highlighted red with yellow descriptive labels. On-screen instructions and help screens are available for each program. To further aid the user, many of the input fields accept a "?", which brings up a selection list of acceptable entries. Clicking the mouse on the label of such fields also brings up the list. All "clickable" labels are color-coded white so that users know where selection lists are available. Thorough SCL error-trapping routines return the cursor to any field with unacceptable input, with appropriate error messages explaining the problem.

![Figure 3](image-url)

**Figure 3**

**Figure 4**

When all input has been correctly entered, the program can be executed. This is done by advancing the cursor to the "LDAW line." The LDAW line is a single character field that accepts execution commands. To execute the program, the user must enter either an 'I' for immediate execution or a 'B' for batch mode execution. Entering an 'I' immediately submits the program for SAS execution. A small
window appears informing the user that the program is running. Upon completion, the user is informed of the success or failure of that run of the program. Entering a 'B' copies the SAS program with the current input parameters to a batch file that can be executed later using the 'Batch' main menu option. Batch programs can also be executed non-interactively from DOS by allowing memory-intensive programs to operate more effectively.

The LDAW line also supports keyboard shortcuts. These shortcuts enable quick maneuvering among menus and submenus. Each main menu option can be selected with a shortcut by typing its first letter on the line. Doing so cancels the current program and proceeds directly to the appropriate submenu corresponding to the main menu option chosen. Using a keyboard shortcut eliminates the need to back out to the main menu in order to access a program under a different submenu. Certain basic system operations, such as setting the libname, are so frequently performed that their execution has also been made possible through keyboard shortcuts. A list of available shortcuts can be viewed by typing a '?' on the LDAW line.

Keyboard shortcuts augment the menu system and can be quite useful. If the user is entering input for an analysis program and wishes to define or change a libname, a keyboard shortcut provides convenient access to the Set Libname program. Typing the shortcut for Set Libname ('L') and pressing 'ENTER' immediately transfers the user to the Set Libname input screen. After modifying the libname, the user needs only to press 'ESC' to exit directly back to the analysis program input screen. Parameters input on that screen by the user prior to recognizing the libname problem reappear immediately upon returning.

Without keyboard shortcuts, the user would have to back out first to the submenu and next to the main menu, and from there select the System submenu followed by the Set Libname option. After changing the path for the libname, the user would have to repeat the above steps in reverse to return to the analysis program. Keyboard shortcuts provide a means for experienced users to further accelerate movement from one program to another.

Analysis Programs

The LDAW features a modular software design approach. The user interface serves as a front-end to a series of modular external SAS analysis programs. When the user executes a program from the LDAW line, input parameters from the user interface are passed into global macro variables using the SCL Submit Block. These macro variables can be referenced by the relevant external SAS program, which in turn is called with a %include statement. This design structure permits the SAS code within analysis programs to be modified as frequently as necessary without the need to recompile the user interface SCL code. This is a special benefit if analysts are proficient in base SAS but not in SAS/AF.

The analysis programs provided with the LDAW perform a variety of tasks, which are summarized in Figure 5. These tasks include inputting data into SAS datasets, performing data validation, analyzing data, and creating presentation-quality graphics of results.

The LDAW System submenu contains options designed to support analysis programs. This group includes programs that manage SAS libnames, set LDAW system defaults, and access the Batch execution system. In addition, a special 'DOS' program allows the user to execute certain common DOS operating system commands without leaving the LDAW.

The LDAW provides considerable flexibility for accessing user data. The user does not need to convert input datasets to an LDAW format. Rather, the LDAW can adapt to different types of user data through a simple mapping table, a Data Description. Data descriptions map LDAW system macro variables, which are referenced by all analysis programs, to corresponding variable names in a user's SAS dataset. System menu options allow the user to create, edit, and delete Data Descriptions. The current Data Description can be changed whenever different types of data are accessed.

Most LDAW data consists of time series information for a specific type of engineering measurement. These data can represent electricity usage, temperature, humidity, flow rates, voltage fluctuation, or virtually any measurement that can be accurately recorded using existing technology. Data are usually gathered through multi-channel monitoring systems that collect data at regular intervals from a given site. Data from all sites are periodically downloaded to a host microcomputer through a modem, and later consolidated into a monthly raw input file. For example, each observation in the input file might consist of 96 15-minute interval readings for one site, for one day, for one channel. Each site often contains four or more channels, representing different aspects and engineering characteristics of the particular site. It is not uncommon for data to be collected from 100 or more sites with multiple end-uses for over one year. This can result in database sizes of several hundred megabytes.

The Read submenu allows users to convert input ASCII files into SAS datasets. The Data Description convention permits a wide variety of user-specified input formats to be supported. In addition, mainframe SAS datasets can be downloaded to the PC. All LDAW analysis programs operate on SAS datasets, and the Read submenu provides a convenient way to translate data into SAS datasets.

The Validate submenu consists of programs that allow the user to edit and validate input SAS datasets. One such program is the Spike Detector, which performs an exhaustive search for data values above a specified cutoff point. This facilitates identification of outliers and bad data readings that inevitably occur. Many data collection systems. The Data Aggregation program converts time series data represented in an, 15, or 30 minute format to 15, 30, or 60 minute format. In addition, the Simple Math procedure permits mathematical operations to be performed on time series data. The Data Cross Validation program compares the sum of end-use channels against a specified whole premise channel. This allows the internal integrity of the data to be verified.

The Analyze submenu consists of programs for smoothing, averaging, and generating energy usage summary statistics. The Smooth option is an implementation of a kernel smoother, designed to reduce variance before averaging. The Daytypes program adds user-defined daytypes to the input SAS dataset for daytype stratification. The averaging procedures can then compute stratified weighted averages of the smoothed data.

The Output submenu generates output files that provide support for spreadsheet programs such as Lotus®, 1-2-3®, and dBASE®, as well as output formats for EPRI products. This allows LDAW analysis results to be exported to support other engineering software and to provide input for related project activities conducted with different database programs.

The Graph submenu produces SAS/GRAPH output of results. The output can be sent to a high-resolution inkjet monitor, or to any SAS supported device. The Load Profile option graphs time of use load profiles. These can be either single day load profiles for a specific site and channel, or average load profiles for an entire channel or site type. Several load profiles can be overlaid on the same graph for comparison. Bar charts can be displayed using the Duty Cycle Distribution program. Pie charts can be generated using the Energy Usage options.

Because users need vary considerably, the LDAW menu system features a separate submenu for customized user analysis modules. Analysis programs written in SAS by the user can be incorporated into the LDAW from this submenu. This can be done interactively without needing to recompile the SCL code.

Using an Undocumented Feature

Program input screens and menus within the LDAW are supported by SCL code. The current implementation of this language does not support user-defined functions which can be called from more than one program. This can make upgrades and maintenance difficult when an application contains many program entries. To avoid such problems, we have incorporated an undocumented feature of SAS/AF which allows general SCL macros to be called within an SCL program entry. The macros must be loaded into RAM before compiling. The use of SCL macros has modularized the LDAW user interface code itself since most changes can now be done in a single place, the SCL macro. This makes it much easier to maintain and to expand the user interface.
Figure 5

**LDAW Analysis Programs**

**System**
- Set Data Description
- Edit a Data Description
- List Datasets
- Set Defaults
- DOS Libraries and Archiving
- Set Libname

**Read**
- Read Flat File
- Modify Dataset
- Descriptive Statistics
- Download Mainframe SSD
- Set Libname

**Validate**
- Simple Math
- Spike Testing
- Duty Cycle Testing
- Indexing System
- Aggregation Module
- Data Cross Validation
- Set Libname

**Analyze**
- Smoother
- Daytypes
- Average by Premise
- Average Across Premises
- Analysis Weights
- Onpeak/Offpeak
- Maximum Usage
- Total Usage
- Set Libname

**User**
- User Modules 1 - 5
- Set Libname

**Graph**
- Load Profiles
- Duty Cycle Histogram
- Connected Loads
- Energy Usage Graphs
- Set Libname

**Output**
- Lotus File
- LDOT File
- RELOAD File
- DSMonitor File
- DBASE File
- Set Libname

**Batch**
- Execute Batch Job
**LDAW Under Other Operating Systems**

The SAS Institute's Multi-Engine Architecture® allows code to be ported between any operating systems that support Version 6 SAS. This feature was recently tested in a transport of SAS/PC® LDAW code to SUN-OS UNIX. The porting procedure is accomplished through PROC CPORT, which creates ASCII transport files. The transport files can then be converted into SAS datasets and catalogs under the relevant operating system using PROC CIMPORT. The porting process worked very smoothly, although some changes had to be made to reconcile DOS and UNIX file systems.

**Conclusion**

The LDAW integrates several SAS software products into an innovative, user-friendly information system. Awkward menu maneuvering found in many SAS/AF applications has been replaced with a smooth, efficient user interface. Options are arranged on and chosen from menus and submenus in a manner familiar to most computer users. The SCL code that supports the user interface has been made much easier to maintain and modify by the use of compiled macros, an undocumented SAS/AF feature. Modularity has been achieved through the use of calls to external SAS analysis programs as well as through extensive use of SCL macros. The end result is a menu-driven software application that makes it possible for users who are not proficient in either SAS/AF or base SAS to perform a full array of analysis programs on SAS datasets.

For more information about the LDAW design, please contact Burt Blodgett or John Powers at:

Quantum Consulting Inc.
2030 Addison Street, Suite 410
Berkeley, CA 94704
415/540-7200


LOTUS and 1-2-3 are the registered trademarks of the Lotus Development Corporation, Cambridge, MA. dBASE is a registered trademark of Ashton-Tate. Macintosh is a registered trademark of Apple Computer, Inc., Cupertino, CA.