Notes From SUGI 23

Jack Hamilton

First Health
West Sacramento, California

Updated with information from SUGI 24

Abstract

These are my notes from SUGI 23, the twenty-third annual SAS Users Group International conference, which was held in Nashville Tennessee, March 22-25, 1998. Most of these notes are specific to Version 7, which is currently in beta. It is expected that production version will be released during the fourth quarter of 1998.

Keywords

SUGI, Version 7, data step, dataset, SQL, Intelligent Storage, Intelligent Servers, Intelligent Clients, Enterprise Guide, Enterprise Reporter, HTML, Output Delivery System

Trademarks

Overview

The logo SAS Institute has been using recently is an inverted pyramid. With Version 7, they have started using three inverted pyramids, each representing a particular aspect of the new SAS system. The three pyramids represent Intelligent Storage, Intelligent Clients, and Intelligent Servers.

A good overview of these three areas, plus changes in the communications infrastructure, was given in the Futures Forum session moderated by Barrett Joyner.

To me, the most significant improvements were the client-side interfaces (including the syntax editor) and "Intelligent Storage". Longer variable names will be useful, and there are lots of things to make statisticians and data analysts happy.

All in all, SAS has done a very good job in designing V7. I can't think of many things that I would have done differently, and those things would be additions rather than replacements. Several times over the course of the conference, developers mentioned design decisions they had to make, and I think they always made the right decision (something which I can't say about Version 6).

SI seems to anticipate what the market wants, without becoming entirely market-driven (some of the new statistical stuff can't possibly pay for itself, but it's there).

Unless otherwise noted, the information below applies only to Version 7.

Intelligent Storage

The Intelligent Storage overview was given by Robert Cross. The major features he described were:

- Long variable names (up to 32 characters), long dataset names (up to 32 characters, with some operating system restrictions), and long character variables (up to 32K bytes). Librefs, filerefs, and format names remain a maximum of 8 characters long.

- Datasets larger than 2 gigabytes. [at least under some operating systems. I'm not sure whether this support is OS-dependent.]

- Data integrity constraints. You can specify checks against values, other variables in the same observation, or values in other datasets. Syntax is similar to that in standard SQL.

- Support for versioning. You can have multiple versions of the same dataset, and refer to a particular generation. This is very similar to GDG's under MVS.

- Transparent cross-environment datasets. You can create a dataset under MVS, transfer it to Unix, and read it directly, without creating an intermediate transport dataset. Access on the receiving platform is read-only. It is also possible to create a dataset in another platform's native format. [Some of these capabilities may require that SAS/CONNECT be licensed.]

- There are more open access tools for ODBC, JDBC, etc. Some of these are available in 6.12 as well.

- Several improvements in SAS/ACCESS: dynamic engines via libname statements, better joins and use of indexes, support for ANSI quoted names (anyone know what that means?), updateable single-table views.

- Various enhancements to SPDS, MDDB, and some other things we don't have licensed.

- Common Metadata repository, object oriented with inheritance and subclassing.

- Hybrid OLAP. I don't remember the details.

Intelligent Servers

"Intelligent Servers" are what do the work (procedures, data storage, etc.), as opposed to clients, which provide the user interface. In the past versions of SAS we all know and love, the client and the server were usually the same (I don't count a terminal or terminal emulator as a intelligent client). With the advent of the World Wide Web, networks, and so forth, the server and the client no longer have to be the same, and there are some advantages to separating them. This overview was given by Alan Eaton.

Data Analysis

- Generalized Estimating Equations for correlated data.

- Exact tests for skewed or sparse data.

- Better confidence interval information in several places.

- New survey data analysis procedures. I talked to someone who's interested in survey research, and he was very excited about what he had heard.

- New nonparametric methods.

- Partial least squares analysis.

- Analyst GUI interface for basic statistical analysis (production in 6.12).

- More stuff in SAS/Insight.

- OR performance improvements.

Business Analysis

- Risk Analysis Engine (Risk Product). Measures market and credit list. Currently a standalone product.

- Enterprise Miner. Has a nice GUI. PROC DMINE for data reduction, various options for neural networks, association, clustering, etc.

Output Delivery System

- The Output Delivery System, or ODS, is a different way of producing output. Currently, all procedures contain code to format output and write it to the output file. ODS separates the report data from the report format. This allows much more control over appearance, and also gives access in the data step to all information calculated by procedures.

- **This might be the single biggest and most obvious improvement in Version 7**.

- I think it's more significant than the increased variable name size and length changes. ODS is described in more detail in a later section of this report.

- They established an "output junta" to review the output from all procedures. The result is a general improvement in quality and consistency across the whole product line.

Reporting

- HTML styles are built in to TABULATE and REPORT, and are available to all procedures through ODS and PROC TEMPLATE.

- There are formatting extensions which allow "traffic lighting" (color highlighting of cells of particular interest) and URL links.

Client Interfaces

- This is mostly discussed in the clients section, but there are changes on the server side to support Enterprise Guide,, Enterprise Reporter, and other client tools.

Intelligent Clients

Deva Kumar talked about intelligent clients, which were the real crowd-pleaser at the opening session. Clients are usually PC's or workstations, but might be web browsers on any platform, or a mainframe with a terminal or terminal emulator.

SAS-Centric Solutions

- These are what most of us use now. They're products like Assist, Analyst, EIS, and so forth. I suppose the DMS, AF, FSP, etc. also count as SAS-Centric solutions. They require SAS to be running on the client.

Windows-Centric Solutions

SAS has created several Windows-specific clients. This is a reasonable thing to do - Windows is the most commonly used desktop platform, and programs written specifically for it in C, or using ActiveX components, will probably be faster and friendlier than more generic applications written in Java.

- Enterprise Guide is a full interface to SAS. It communicates with a server, and provides access to data on both the PC and the server. There are several ways to display projects and information. You can show all the data and programs associated with a project. You can show all the data associated with a particular data source. Transparent data access has been a major theme in developing V7, and one thing they demonstrated at the opening session impressed me greatly: they clicked on a data icon in the project tree and dragged it to another part of the tree. Underneath, SAS read an Excel dataset on the PC and copied it to a SAS dataset on a server. Quite impressive. [I might have the details wrong, but it was something close.] You can also write code and submit it to the server. Enterprise Guide doesn't require that SAS be running on the PC.

- Enterprise Reporter is a stand-alone report writing tool. It provides point-and-click report-writing capabilities.

- The Enhanced Editor knows about SAS syntax, so it highlights keywords, changes the color of quoted text, and so forth. It's in SAS for Windows and Enterprise Guide. Everyone who saw it was very enthusiastic about the editor. It will help prevent a lot of careless coding errors.

Web-Centric Solutions

- There's an HTML-based MDDB viewer (if it's what I'm thinking of from another session, it allows drill-downs and other neat stuff).

- ~~At some previous conference, I thought I saw something similar to Enterprise Reporter, but written in Java. This would certainly be possible, given the ability to read data with JDBC and submit remote jobs with JConnect. This would probably be slower than the pure Windows solution, since a web server would be in the middle, but it would provide a lot of power in a fairly platform-independent implementation.~~ **SUGI24:** This seems to have disappeared.

Communications Infrastructure

Jack Wallace talked about some changes to the communications infrastructure:

SAS Client/Server Interfaces

Asynchronous RSUBMIT

- Messaging and Message Queuing, including the ability to send attachments and to create workflow applications. It will be possible queue a message for an application that isn't running, and have the message processed when the application becomes available.

- Remote Objects

- Agent services, allowing programs to be run on schedule

- Network Data Encryption.

- Open Client/Server Interfaces

- Data access interfaces through ODBC, JDBC, OLE/DB, SQL library.

- Code Submission interface through OLE Automation, CGI broker, and JConnect. Some of this is already in 6.12.

- Object interfaces (prototype only in V7).

  - System objects for data access, code submission, output retrieval, user written objects.

  - Object transports include COM/DCOM, CORBA, and JConnect.

Other Items from the Futures Forum

There was a question and answer session after all the speakers had given their presentations. Some of the Q&A:

Q: Will SAS support Unicode?

A: Maybe someday. Not in the first release on all platforms. Unicode support isn't standardized in Unix.

Q: Will there be a Version 7 on the Macintosh?

A: It doesn't seem to be cost-effective to port it. Porting is expensive, and there's not enough demand to justify it.

Q: Will there be a Version 7 for Linux?

A: Again, there doesn't seem to be the demand to justify it. [It seemed to me that more people want a Linus version than a Macintosh version, but I have no idea whether the audience was representative.]

Q: You mentioned an "output junta" which reviewed output for consistency. I hope you had or will have a "code junta" which has the goal of eliminating all the little inconsistencies which have crept into the language over the years. [This was my question.]

A: Would you want this if it meant that some existing programs would break? It might might mean that. [My answer: yes, I would. I expect that some programs will break anyway. Increased consistency would make writing new programs so much easier that a few startup problems are worth the price. In any case, it might not be necessary to make old syntax go away in order to add new, consistent syntax.]

Data Step Changes

- _FILE_ and _INFILE_ pseudo variables give access to current file buffers. They can be used pretty much everywhere regular variables can be used. I've been wanting something like that for a long time. I found it very useful in PL/I.

- DATASTMTCHK system option to signal an error if certain data step keywords are used in the DATA statement. This is specifically to prevent accidentally erasing an input dataset when you forget a semicolon, as in

```
data new
   set keepme.bigdata;
```

In V6, this would cause datasets named SET and KEEPME.BIGDATA to be created, erasing the old KEEPME.BIGDATA in the process. With this new option, having a dataset named SET (or MERGE, etc.) will cause an error. You can turn this option off, or make it check for a large or small set of keywords.

- FOOTNOTEs are now supported in data step output.

- The DSD (delimiter sensitive data) option is now supported for the FILE statement as well as the INFILE statement.

- Some optimization of data step code will be done - storing values in registers, etc.

Dataset Changes

- Common extension, SAS7BDAT, on all platforms.

- Regular datasets are shareable, read-only, across all platforms. No more need for CPORT and CIMPORT! [Probably will require SAS/CONNECT license on reading side.] Indexes won't work cross-platform, nor will formats.

- Only one transport dataset format, the CPORT format, which can store datasets and catalogs.

- Can refer to datasets by quoted name, e.g.

```
DATA
  'd:\sasdata\newstuff.sas7bdat';
                                        SET
  'd:\sasdata\oldstuff.sas7bdat';
```

- Better compression, including compression of numerics. Compressed datasets can now be accessed by observation number. A compressed dataset will never be larger than the uncompressed version would have been. Compression is still at the observation level. There's a user-written compression option if you know your data well.

- Variable and dataset names can be up to 32 characters long, in mixed case, and can even include special characters (there's a new name constant to make this possible - "name with spaces"n). Using non-alphanumeric characters will probably break some existing programs, so there's a system option to control the degree of naming freedom you have.

- Labels can be up to 256 bytes long.

- Transparent data access will let you read data in a database without creating a SAS/Access view first. This is made much easier by the longer names.

- Can concatenate libraries and catalogs. The concatenated libraries can have different engines. You can have all your data appear in one folder in the SAS Explorer, even though the underlying data are in many different locations.

- Versions are supported, using the new dataset options GENMAX= and GENNUM=. Modeled after GDG's on MVS.

- PROC APPEND is much more efficient when appending to an indexed dataset.

- Integrity constraints restrict the values that be inserted into a dataset. This can be done by applications now, but there's no guarantee that all applications will use the same (or any) code. The new constraints apply to any write access to the dataset. Standard ANSI SQL constraints are supported.

- You can create updateable SQL views. A view can update only one underlying table.

- Indexes store information on the distribution of values. In general, better use is made of indexes, and you have more control when the system makes the wrong decision.

Windows Clients

- Can increase the size of the command line, and can recall up to 99 commands.

- Saves settings on exit.

- Better AutoComplete of commands.

- Preferences dialog will be much better organized.

- Supports the Microsoft IntelliMouse.

- Better Print Preview.

- Print enhancements - standard dialogs for page setup and printing, better batch printing, better ways to set margins and orientation.

- Rectangular area selection with Alt-Click [also in v6 - I don't know that].

- Can save in Microsoft Rich Text Format [this is done through the Output Delivery System, so it probably applies to all platforms]. **SUGI24:** RTF is experimental in version 7.

- File filters in file dialogs.

- "Designed for Microsoft BackOffice" logo, which means that it is network independent, supports NT security, can run as an NT service, and can use MS Systems Management Support for automatic remote or local installation.

- Requires Windows95, Windows98, Nt4, or NT5. For some services, it also requires that Internet Explorer be installed, but you won't have to use IE as your browser.

- The SAS Universal ODBC Driver (available now for v6) lets an ODBC client such as Excel read SAS data without having SAS installed on the PC. Access is read-only. This product is sold separately, and costs $99 (less per user for multiple-user licenses). You can download a 30-day evaluation copy.

- The SAS System Viewer, also available now for v6 (free!), is a standalone program which reads SAS datasets. You can't save the data directly, but you can copy it to the clipboard and then insert it into another program, such as Excel. The number of observations you can read is limited, and the viewer cannot be controlled programmatically.

All Clients

- Filenames and filetypes are now tracked per window, making it more difficult to accidentally overwrite your program file with the log file. [This was mentioned in a Windows session, but I think it applies to all platforms.]

- Autosave on all platforms.

- Help now uses HTML format, which makes local changes easier. [This was mentioned in a Windows session, but I think it applies to all platforms.]

Output Delivery System

- The Output Delivery System separates the creation of data by a PROC from the presentation of that data. This makes procedures smaller and easier to write.

- You can have much more control over many aspects of your printout. For example, you can specify the format and the number of decimal places for numeric data.

- You can get programmatic access to all procedure output. You no longer have to read and parse a print file in order to obtain all the numbers in statistical procedures.

- You can run a procedure once and display the output in several different formats. It's easier to save results for later use or display.

- You can insure a common style for all output from a job or set of jobs.

- There was some disagreement among the speakers in various sessions about what output formats would be supported, and when, but plain text output and HTML will be available in the first release, with PCL and PS formatting available either then or in a subsequent release. PDF and LaTex are under consideration. **SUGI24:** PCL, PDF, LaTex, and plain text are not supported in version 7.

Miscellaneous

- New option for unbuffered log output, so the entire log will be available in case of an abend.

- Macro names can be up to 32 characters long, but names in autocall libraries are subject to operating system restrictions.

- PROC FORMAT has enhancements for creating date formats~~ and informats~~.

Suggestions

I always take advantage of the access to SAS developers that SUGI gives me by making suggestions. Here's my list from SUGI 23. They're of varying levels of importance, and some of them have not been completely thought through. They're not in any particular order.

If you agree with any of these suggestions, please let SAS Technical Support and your sales rep know. If you don't agree, uh, I guess not saying anything is one way way to express your disagreement. I will write up more information about these upon request.

- Add a system option to make SAS emphasize resource usage rather than appearance in X-Windows. In V6, the Motif interface under VMS creates an unreasonable amount of network traffic. I don't need beveled windows if every OK box takes 15 seconds to load.

- Make file size available through the data step information functions on all platforms. Make LRECL, RECFM, and Owner available whenever possible. In V7, the SQL constraint information should be available.

- Add an option on INFILE and FILENAME which specifies the record separator. I occasionally have to read files which, for whatever reason, don't contain the standard EOL for the operating system, and it can be incredibly difficult for a task which ought to be so simple. The SOCKET interface has a TERMSTR= option which might fit the bill, but it should be available for all types of input.

- Support VTxxx-style terminal emulation under OpenVMS. This is currently scheduled to go away in Version 7. Apparently 3270 terminals will continue to be supported under MVS, which is good news for users on that platform. **SUGI24:** VTxxx support probably won't happen ever.

- Provide a way to specify that all versions of versioned datasets be concatenated and read at once.

- Provide a mechanism for copying attributes from a previous generation when creating a new version of a versioned dataset. [I had some discussion with one of the developers about this, and have been thinking about it, and a general solution is not as straightforward as I first thought.]

- Provide better explanations of items on the SASWare Ballot.

- Provide an informat that attempts to convert any kind of date string to a date value, similar to what Excel does.

- Provide functions and formats to handle the new quoted name construct. This will make it easier to write applications that don't break on datasets with new-style names.

- Create boolean shortcut comparison operators.

- Support "virtual" variables in datasets. The value of a virtual variable would be calculated when it's accessed, rather than stored in the dataset. This would save storage space when a value depends entirely on other variables in the same observation. Example (not necessarily a good one): a variable WeekDay which is just the day of the week calculated from a date value in another variable. This could be indexed, making it possible to read all the values for Tuesdays without running the WEEKDAY function on every observation.

- Enhance PROC FORMAT to provide a syntax for specifying the format name and other format information as constants when using a CNTLIN dataset that doesn't contain all the needed information, and allow the creation of formats which reference values in a dataset, rather than storing values in the format.