

Easy Interface for SAS® -- A SAS® Application on the Windows® Platform

Zhuan (John) Xu, 1ST Consulting L.L.C., West Des Moines, Iowa

ABSTRACT

Easy Interface for SAS is a Windows application that provides a user-friendly graphical user interface to the SAS system. Users are provided with a main screen styled after Windows Explorer. EI allows users to summarize data, calculate correlations, and perform sample size estimation, random sampling, T-tests, ANOVA and regression. In addition, EI supports the viewing, subsetting, sorting, merging, importing and exporting of SAS data sets. Its DOLAP (Desktop Online Analytic Processing) function is powerful yet quite easy to use for slicing and dicing and drilling down. Another nice feature is EIS Ad Hoc Application, which allows most SAS programs to be integrated into Easy Interface for SAS by creating a dynamic GUI.

INTRODUCTION

Easy Interface for SAS® (EI) is a Windows® Application on PC platform. It adds an easy-to-use graphic user interface to SAS software through OLE automation. The technical details of developing such application can be found from the SAS on-line documentation about OLE automation and from the author's SUGI23 paper¹ titled 'Using SAS® as an Automation Server in Windows Application Development'.

EI is designed as an exploratory data analysis tool. With EI, users can perform data management, data and statistical analysis interactively. As the use of EI does not require SAS programming skill, people who know the business side can use EI to do various analyses without formal training in SAS programming. As the users have access to the SAS program generated, EI can be used to learn SAS programming as well. More advanced functionality, such as Desktop OLAP and Ad Hoc Application are very useful to all level of SAS users.

ARCHITECTURE & TECHNICAL BACKGROUD

Easy Interface for SAS® uses SAS OLE Automation Server as a database and analytic engine. It acts as an automation client to control the SAS server. This technology is widely used in Windows applications. In addition to SAS, most major Windows applications, such as Microsoft Word, Excel, etc. are automation servers. Many Application development tools, such as Visual Basic, Visual C++, Delphi, and SAS/AF support OLE automation.

Conceptually, EI is quite simple. It accepts user input to generate a SAS program. The resulting SAS program is then submitted to the SAS server. After SAS finishes processing EI retrieves the result. The result may be displayed immediately or be processed further. For example, EI may use the text output from SAS to produce a bar chart. Figure 1 shows how SAS interacts with EI.

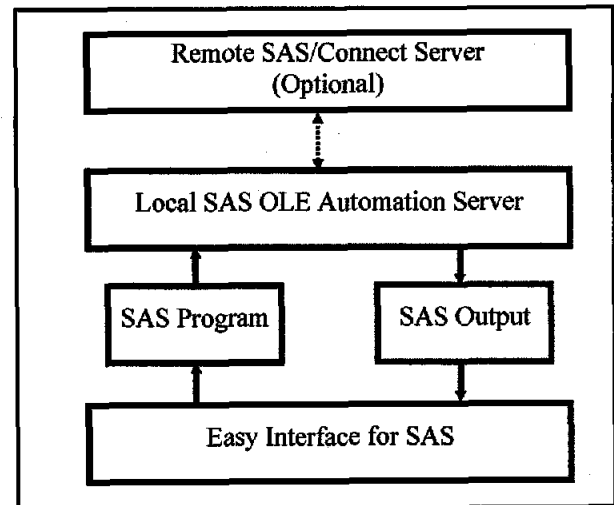


Figure 1 - SAS and Easy Interface for SAS

The basic structure of EI can be illustrated by Figure 2. In Data Manager, the user performs various file management tasks and selects a procedure to use. In Procedure, the user sets up parameters and EI generates a SAS program based on user input. The SAS program is then submitted to SAS OLE server and the result is displayed in the Output Window. The actual implementation is more complicated since the interaction between the SAS server and front-end is limited.

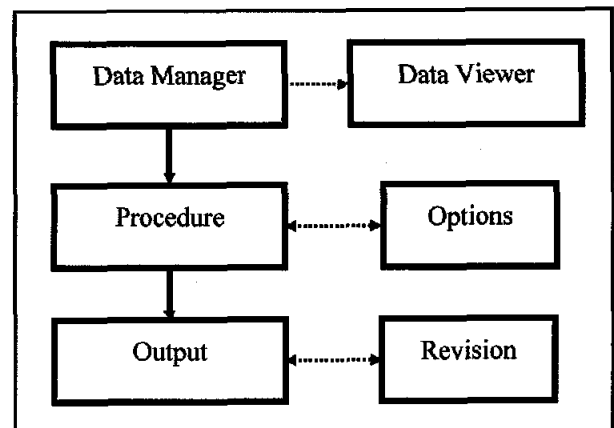


Figure 2 - Basic Structure of Easy Interface for SAS

Selecting a development tool is a very important decision. The author compared Microsoft Visual Basic (VB), Visual C++, and Inprise (Borland) Delphi in the process. Although Delphi is less popular than other two in the market, it is a Pascal-based object-oriented development tool as easy as VB and as powerful as C++. The resulting application runs much faster than VB and is almost as fast as C++. This makes it possible to add more features to EI with limited resources and time. Additionally, it produces only one application file so the distribution and installation can be simplified (and while the current version of EI

includes a DLL file to support a local database in Desktop OLAP, all other functionality works fine without it).

USER INTERFACE DESIGN

As the name implies, EI is designed to be easy to use. The author designed the interface to be:

- Graphical
- Simple
- Familiar
- Flexible

Most Windows development tools create graphical user interfaces. Even so, attention is paid to minimize the user input in order to reduce potential error. For example, in EI users always select a variable using a button or drag/drop method instead of typing the variable name.

A simple user interface makes the application easier to use. Most procedures in EI have default settings for options. This way, the user can start to use the procedure by selecting only necessary parameters. As the user feels more comfortable he or she can display the options and revise them. As another example, to make it easier for the user to work with SAS data sets, EI will automatically search a user-specified drive or directory and display only those directories containing SAS data.

A graphical user interface is nice but a familiar user interface is even better as it significantly reduces the learning curve. Since most people are familiar with Windows Explorer, the main window of EI, Data Manager, looks and works similarly. For the same reason, the SQL part is designed with a style similar to the Microsoft Access query window. As for the date input, SAS typically uses '23OCT1998'D format. But people are more familiar with date in 10/23/1998 format. EI enables the user to enter dates in the familiar format and then convert them back to a SAS format when generating the SAS program.

MAIN FUNCTIONALITY IMPLEMENTED IN EI

EI's functionality can roughly be classified into three major groups: data management, statistics, and analysis.

Data management functionality includes:

- Copying, moving, and deleting SAS data sets
- Renaming SAS data sets
- Browsing SAS data sets
- Performing SAS Data Steps with support for subsetting, merging, and other SAS statements.
- SQL subsetting
- SQL summary
- Sorting data
- Importing data from text files and Microsoft Excel files
- Exporting SAS data to text files, Microsoft Excel and Word files
- SAS Format management

Statistics functionality includes:

- Summary
- Frequency
- Correlation
- T-Test
- Paired T-Test
- Analysis of Variance (ANOVA)
- Regression
- Confidence intervals for means and proportion
- Sample size estimation for means and proportion with one and two samples
- Fast statistically valid random sampling

Analysis functionality includes:

- Desktop OLAP
- Tabulate Report
- Benchmark analysis
- Histogram
- Ad Hoc Application

The remainder of this paper describes Easy Interface for SAS in more details. The focus is on those functions the author feels are more interesting, more useful, or not easy to do with SAS directly.

DATA EXPLORER

Data Explorer (See Figure 3) is the main window of EI. It is styled after the Windows Explorer with a main menu, a tool bar, a status bar, a directory list, a file list, and a context-sensitive pop-up menu. It not only looks like Windows Explorer but also works similarly. For example, click on the header titled Name in the file list will sort the file by name. As another example, to rename a SAS data set, just select the file and click on the file name. This will activate the in-place editor. Now you can type in a new name. Just make sure it's a valid SAS name with file extension .sd2. You can also drag a file to a new directory. As in Windows Explorer, you can choose to view the file list in Large Icons, Small Icons, List, or Detail views. You can also hide the toolbar or status bar, and resize the window. These settings will be saved by EI and applied next time you start EI.

Obviously, there is no need for Data Explorer if it is identical to Windows Explorer. However, Data Explorer is the launch pad to all EI functionality. You can double click on the SAS file name to browse the data. To start a procedure, you typically select the file then select the corresponding menu item. In addition, EI only displays the file type selected, both in the directory list and in file list. To update the directory list, you pick a drive from the drive/folder box on the tool bar or type in the drive name or a directory name. Next select a file type from the box next to it. Most of the time, the file type should be SAS Dataset (*.sd2). Click on the Search button next to it to start search. EI will search the selected drive or directory for the desired file type, only those directories that contain such files will be displayed in the directory list. As the number of directories containing SAS data should not be too many, (or you better get organized), a one-level

list is used instead of the tree structure in Windows Explorer. Clicking on a directory will display the files in the directory on the right. Finally, you can turn on the library mode, which shows SAS libraries on the left-hand side and SAS data sets on the right. This mode is necessary to access the remote data through SAS/CONNECT.

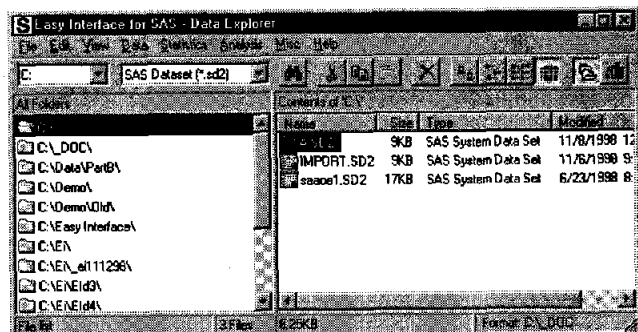


Figure 3 - Data Explorer

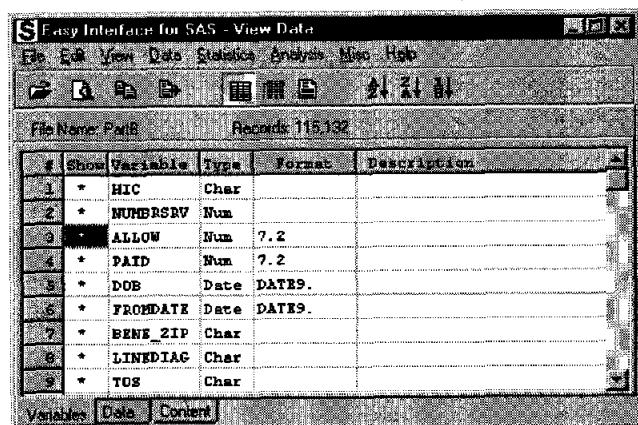


Figure 4 - Data Viewer showing the variable list

DATA VIEWER

Many SAS programmers just use PROC PRINT to view data. Some use FSVIEW or the newer VIEWTABLE. EI has a built-in data viewer intended to make browsing data more user friendly. Double clicking on a SAS data file name in Data Explorer will start Data Viewer. It actually provides three views to the data selected. The first one is a list showing variable names and related information such as variable type (Numeric, Character, or Date), Format, variable description (Label) in a table (See Figure 4). The second is the data in a table format (See Figure 5). The third view is the output of PROC CONTENTS. You need to click on the tab at the bottom to switch among different views.

The variable list is also used to configure the data view. Clicking once in a cell below the Show column will hide the asterisk (*). In turn, this variable will disappear from the data view. Clicking the same cell will return the asterisk (*) to the cell. You can click any of the Sort buttons to sort

the variable list by the column in which the active cell is displayed. The default is the SAS internal order. You can also change the order by dragging any row to a new position. To change the variable's display format, click on Edit | Change Format in the menu. This will turn on the format edit mode. You can now add or change the format displayed.

Any change in the variable view will be reflected in the data view. Additionally, you can drag a column to a new location to arrange the order. For a small data set, EI will display all observations in the data view. You can then sort the data using the buttons on the toolbar. For a larger data set, clicking on the Sort buttons will pop up a new sorting window and then you can have SAS sort the data set.

The Data Viewer provides the ability to print and export through the buttons on the toolbar. Like Data Explorer it can be used as a launch pad for other procedures since they are accessible from the menu at the top.

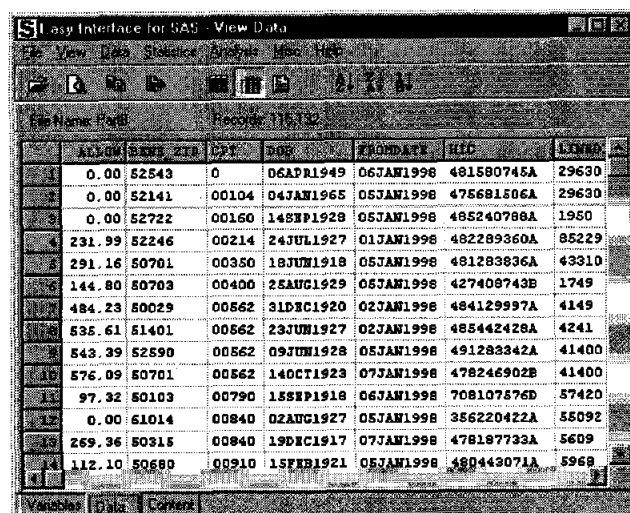


Figure 5 - Data Viewer showing data

DATA MANAGEMENT

Easy Interface for SAS has strong data management capabilities. You can use SQL Subset (shown in Figure 6) to select a subset of data for a new SAS data set, or use SQL Summary (shown in Figure 7) to summarize data for a new SAS set. To specify a join for two data sets, you may open another SAS data set and drag a variable from one on top of another in another (similar to Microsoft Access). The join type can be an inner join or an outer join. You can change the join type easily by clicking on the diagram in the middle. To build an SQL query, you drag a variable to the grid at the lower part of the window or click on a button on the toolbar. To delete a column, click on the top cell of the column. The cell will turn red and show the word 'Delete' (See Figure 7). Now click on the right mouse button to delete that column.

To produce a selection criteria, invoke the Where Clause Builder (as shown in Figure 8) by a double clicking on the cell in the grid. The Where Clause Builder supports many

functions and operators corresponding to the variable type. For example, for a character variable representing phone number, SUBSTR function can be used to get only the area code part of the phone number. You can then select a proper operator to create the desired where clause. Commonly-used operators are supported.

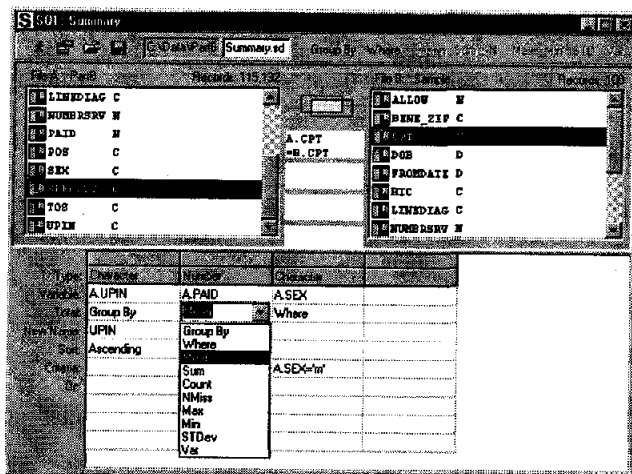


Figure 6 - SQL Summary

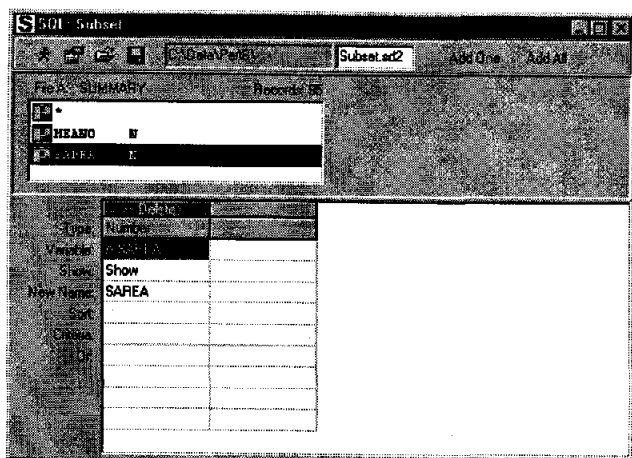


Figure 7 - SQL Subset

El also has the Data Step (see Figure 9) for SAS work typically done in a data step, such as subsetting and merging. Data step has a Sample Code Template containing many commonly used SAS statements. You can use it to help build additional SAS statements in the data step. The Where Clause Builder can also be invoked in Data Step.

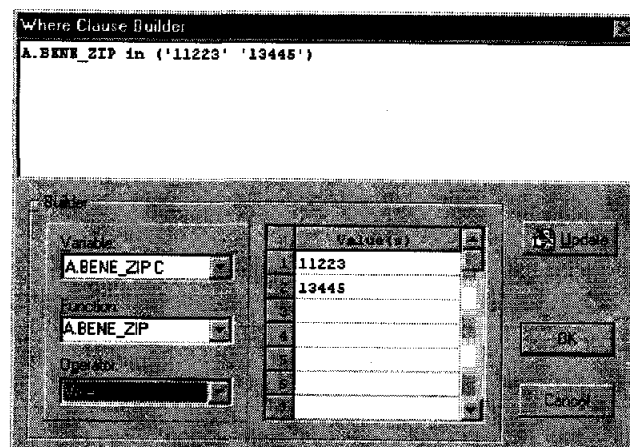


Figure 8 - Where Clause Builder

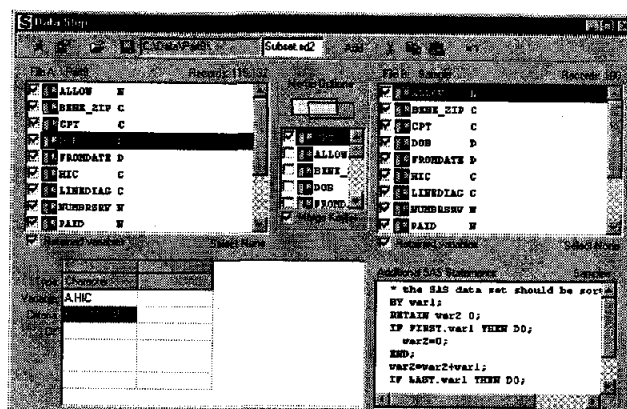


Figure 9 - Data Step

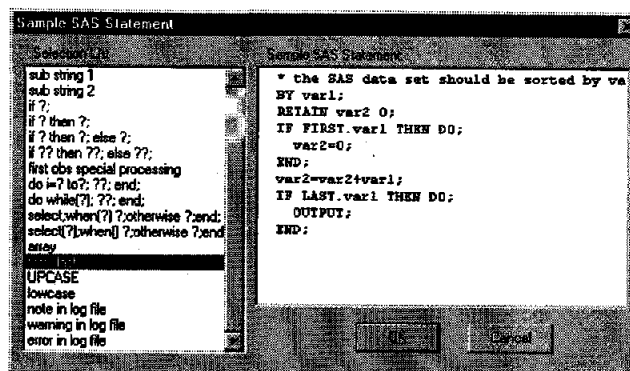


Figure 10 - Sample SAS Statement

EXPORT SAS DATA

Easy Interface for SAS makes exporting SAS data to other formats very simple. Just select a SAS file in Data Explorer and click on the menu Data | Export. You will then get a new window similar to the one in Figure 11. Exclude some variables if you prefer. Otherwise, just click on one of the four buttons on the toolbar to export data to

a tab-delimited text file, comma-delimited text file, Microsoft Excel, or Microsoft Word document. The text file formats require you to enter a file name. For Word, EI will simply start Word and display the data in a new file. EI works with Excel in the same way. Note that this means you need to have Word or Excel installed on your PC in order to export to these formats.

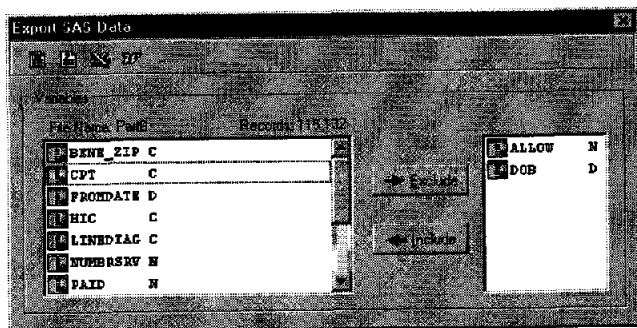


Figure 11 - Sample SAS Statement

		SEX	
		1	2
		PAID	PAID
		SUM	SUM
SPECLTY	D1	7,197.69	7,646.15

Figure 12 - Output Window

STATISTICS

EI supports basic statistics including Descriptive Statistics (PROC MEANS and PROC FREQ), Correlation, T-Test, Paired T-Test, ANOVA, Regression, Confidence Interval, Sample Size Estimates, and Random Sampling. For users other than professional statisticians, these should be sufficient in most cases.

Most of them work in a similar way: you select a SAS data set, click the corresponding menu item to start a new window, select the variables to be analyzed and set options if needed. You may click the Run button or click the Code button to first review the EI-generated SAS program. The output is always displayed in the Output Window (see Figure 12). Note that you can view the SAS program and its log file as well. You may also revise the SAS program and rerun it in the Output Window. There are many features in the Output Window, such as Copy/Cut/Paste/Undo, Find and Replace, text-formatting etc. They are accessible through the menu and toolbar.

For statistics, let's use Summary and Frequency as examples because they are used frequently.

Summary provides a simple user interface to PROC MEANS in SAS/BASE. Users can select analysis and group variables. Statistics are listed in the options (see Figure 13). Mean, Sum, and Standard Deviation are default statistics. All statistics can be selected or deselected based on need. Check Test Mean=0 if desired. The maximum decimal places is set at 2 by default but can be changed.

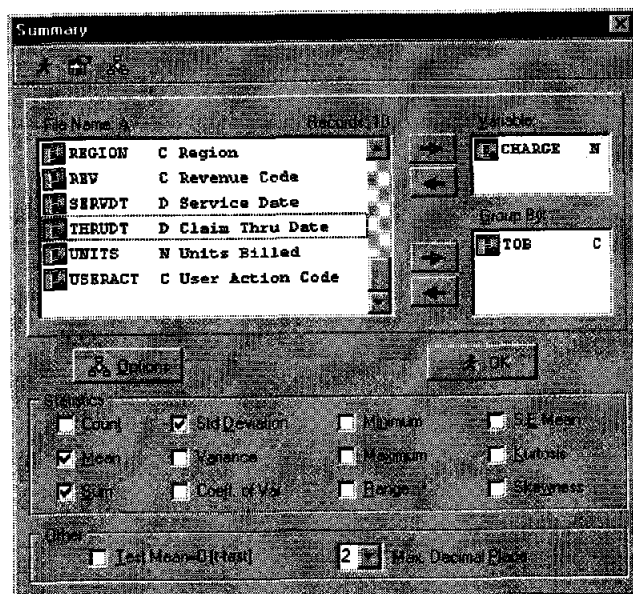


Figure 13 - Summary

Frequency implements the functionality in PROC FREQ. Users can select variables to be counted and set commonly used options. There are three types of options: analysis options allows the user to choose None, Chi-Square, or All statistics. Output options change the display of the result. Other options permits the user to set a weight variable; set the sorting order in the output; and decide if missing values should be included or not (see Figure 14).

As mentioned above, since other statistics work similarly, the details are omitted here. Since sample size estimation does not have a corresponding SAS procedure a sample screen for one sample is shown in Figure 15.

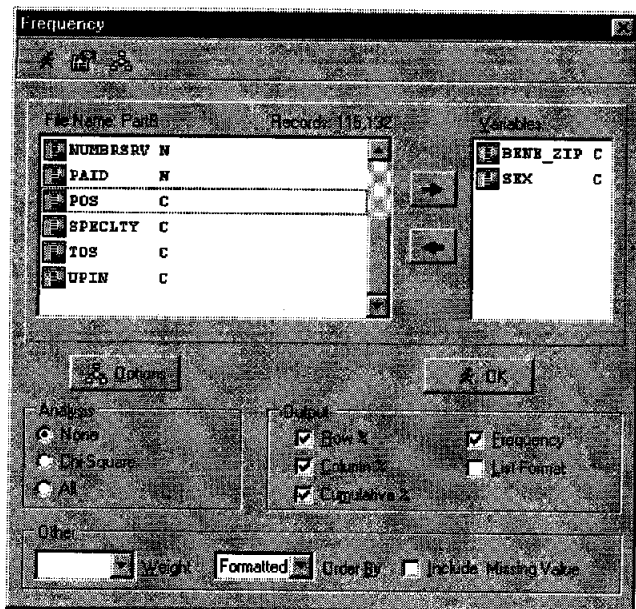


Figure 14 - Frequency

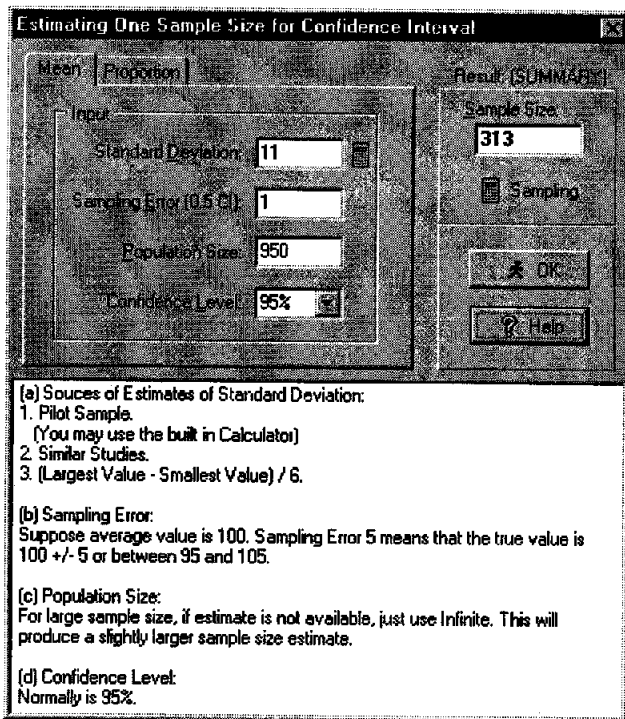


Figure15 - Sample Size Estimating

DESKTOP OLAP

OLAP (Online Analytic Processing) is a popular way for users to look at their data to answer business problems. It provides an easy way to do 'slicing and dicing' and 'drilling down' the data.

As an example, suppose we are interested in the sales performance of a company. We want to see the sales by region and product, as well as trending on time. Furthermore, if we find something interesting, such as the trending showing the overall sales of the current period is better than past periods, we will want to 'drill down' to see further details. We may want to know if certain regions do much better than other regions or if a new product made the difference. This is called 'slicing and dicing'. Note that this is a dynamic process: findings from one step will help us to decide what to look into in the next step.

In general, OLAP is a discovery and query tool for uncovering information as well as a support tool for analysis and confirmation at the overview level.

The SAS Institute Inc. has specialized OLAP products. SAS/MDDDB is a tool for creating a multi-dimensional database. SAS/EIS includes some multi-dimensional objects specially designed for OLAP. One can also use SAS/AF to develop an OLAP front-end. These tools make SAS a good choice for OLAP-type analyses. Details of SAS OLAP tools and their application are beyond this paper and can be found in SAS documentation and at the SAS Web site. You may also find a good discussion in Peter R. Welbrock's book².

It is worth mentioning that PROC MEANS and PROC SUMMARY, two very similar procedures for simple statistics in the SAS/BASE product, provide us with a tool to generate results closely related to OLAP. Using the sales example in the previous section, we may have a SAS data set SALES, which has all sales transaction data, including three dimension variables (REGION, PRODUCT, and TIME) and the analysis variable AMOUNT. The following simple SAS program will produce all the data we need for OLAP:

```
/* summarize data */
proc means noprint data=SALES;
  var AMOUNT;
  class REGION PRODUCT TIME;
  output out=RESULT n=N sum=SUM
    mean=MEAN std=STD;
run;

/* output result */
data _null_;
  set RESULT;
  file 'C:\RESULT.TXT';
  LEVEL=put(_type_,binary3.);
  /* write header */
  if _n_=1 then
    put "LEVEL"      '09'x
       "REGION"     '09'x
       "PRODUCT"    '09'x
       "TIME"       '09'x
       "N"          '09'x
       "MEAN"       '09'x
       "SUM"        '09'x
       "STD"        '09'x
       ;
  put  LEVEL      '09'x
     REGION      '09'x
     PRODUCT     '09'x
     TIME        '09'x
     N           '09'x
     MEAN        '09'x
     SUM         '09'x
     STD         '09'x
```

```
run;
```

The resulting output is a tab-delimited text file similar to the following:

LEVEL	REGION	PRODUCT	TIME	N	MEAN	SUM	STD
000				100	50	5000	40
001			97	40	45	1800	30
.....							
010			1	20	50	1000	40
.....							
100	East			30	40	1200	45
.....							
011			1 97	10	30	300	33
.....							
101	East		97	10	40	400	46
...							
110	East		1	10	45	450	45
...							
111	East		1 97	3	44	132	33
.....							

Note that the use of class statement is the key in generating such multi-dimensional summary. Also note that LEVEL represents the dimension and it is the SAS generated variable _TYPE_ in binary format.

Now to create a OLAP tool, we first need a graphic user interface to select desired analysis variable and dimensional variables. And then the corresponding SAS program should be dynamically generated and submitted to produce the result file. Finally, the most important part is a user friendly front end for us to do 'slicing and dicing' and 'drilling down' the multi-dimensional summary statistics. This front end is implemented in the Easy Interface for SAS.

As discussed earlier, SAS/BASE can be used to generate multi-dimensional summary. The goal is to create a simple yet powerful front end to create and access the summary.

OLAP tools are often used by a knowledge worker, who is more interested in extracting information and may not be skilled in programming or using software. It is desired to have a simple graphic user interface support 'point and click'. In addition, as the user will decide the next step of analysis based on the finding. It would be nice to integrate the user input with output so the user can focus on it. Furthermore, result should be displayed in both numeric and graph.

From programming point of view, building above functionality from ground up is a big task. Fortunately, most modern programming tools provide components ready to use. For example, Borland Delphi comes with database, report, and graph components. These are three major pieces used in OLAP part of EI.

The first part of OLAP in EI was designed to dynamically create multi-dimensional summary based on your selection. The second part is for conducting OLAP. The screen shots are shown in Figure 16 and Figure 17, respectively.

Using the OLAP functionality in EI is quite easy. You first start EI and select a SAS data set to be analyzed with the

mouse.. Then choose Analysis, OLAP, New... from the menu. This will start a new window similar to Figure 16. Now you can choose Analysis (Factual) Variable and Dimensional (Descriptive) Variables by 'Point and Click' or 'Drag and Drop'. Click OK button to start summarization. When it is done, EI will read the summary and display a new window with a chart and a table (See Figure 17). This is the place you actually conduct analysis. You may:

- Slice and dice to another dimension.
- Drill down to next level.
- Select different statistics for analysis.
- Sort the table and chart.
- Save data or graph.
- Print/Preview report/graph.
- Copy data/graph to other application through Windows' Clipboard.
- Change chart among Bar Chart, Pie Chart, and Line Chart.
- Make comparison.
- Change display label, title, and foot note etc.

You mostly use 'point and click' to do the above tasks. For example, clicking on a blank cell in another dimension at the same level with switch to that dimension. This is called 'slicing and dicing'. Click on a blank at a lower level will drill down. Click on a cell of statistics (Count, Average, Total, STD) will change the chart to display the selected statistics. Click on the header will sort the data and chart. For example, click the header titled Total will sort the data and chart by the total of the analysis variable.

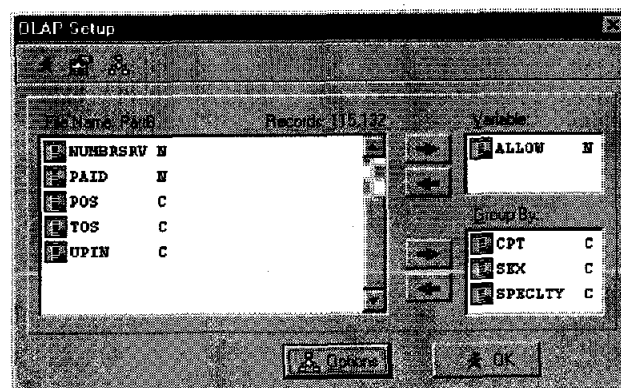


Figure 16 - Desktop OLAP Set Up

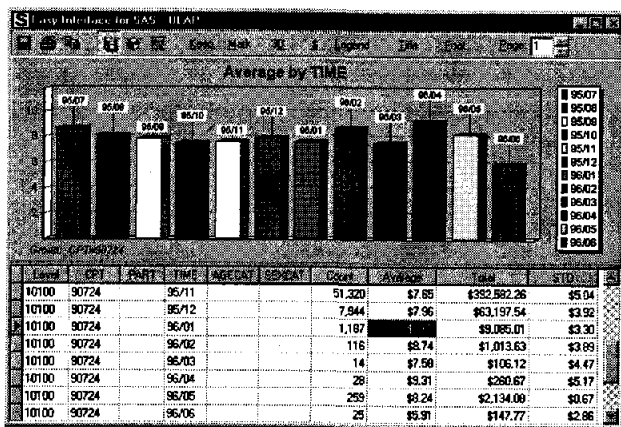


Figure 17 - Desktop OLAP

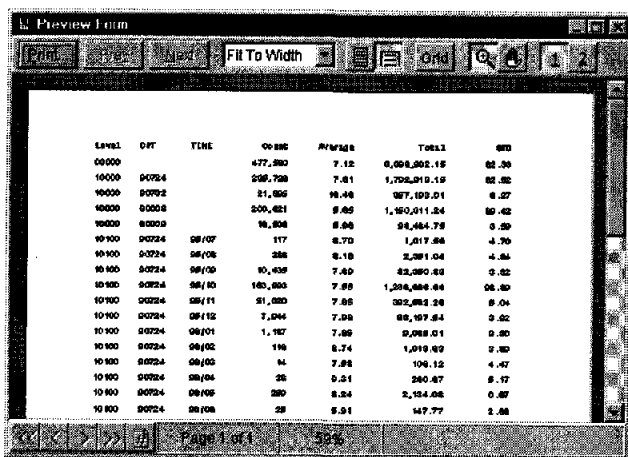


Figure 18 - Desktop OLAP Print Preview

Click on the buttons in the toolbar will let you accomplish other tasks. For example, click on the button with a printer then select Report from the pop up menu will display a report preview window, as show in Figure 18. You can then print the report. If you have saved the summary data, you can simply choose Analysis, Olap, Open... from menu to bypass data summary step. EI also automatically save the last OLAP summary data in c:\temp\olap.dat. So you can always continue the OLAP analysis you did last time. Furthermore, the summary file can be forwarded to another user of EI. In this case, since the summary has been done, the new user can perform OLAP with EI alone, without SAS installed on the computer.

AD HOC APPLICATION

Many times, a SAS program is developed for a business purpose. It is then used again and again with minor changes. SAS language allows us to define macro variables to turn such changes into parameters. One then need only to change the value assigned to the macro variable before rerun the SAS program. To further simplify such task, 'Ad Hoc Application' was implemented in EI. This enables us to add a more meaningful description to the macro variable name. We can also define a pick list for users to select different parameters. The main

advantage is that this allows the end users, especially those not familiar with SAS software to run such SAS programs with a simple user interface to change macro variables, without touching the SAS code at all.

Figure 19 shows the screen for programmer to setup Ad Hoc Application. Figure 20 is the screen for end users to run the application.

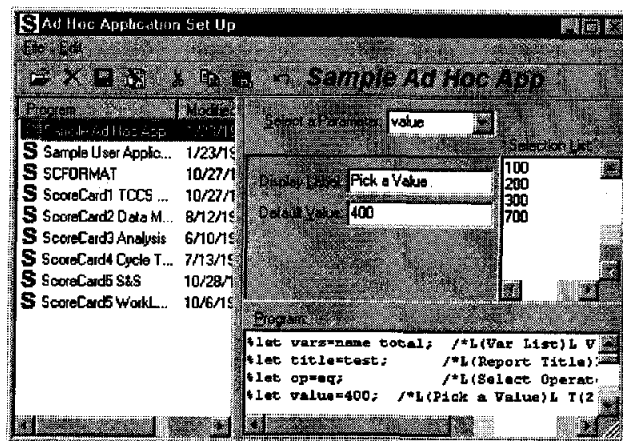


Figure 19 - Ad Hoc Application Set Up

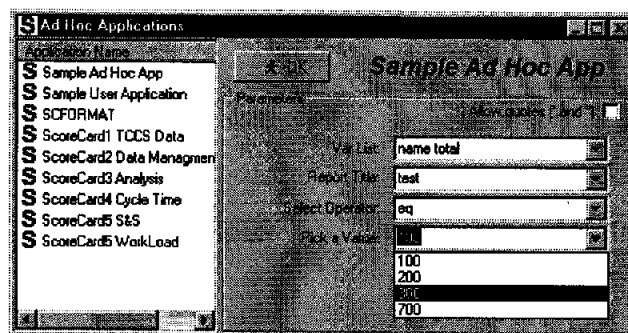


Figure 20 - Ad Hoc Application

Ad Hoc Application supports SAS/Connect. EI will check the output file and display it when the processing is complete.

SAS FORMAT MANAGEMENT

SAS format is a unique feature in SAS software. It can be used to format the output, or as a lookup table, or as a user defined function, or a simple way to re-code variables in a SAS procedure. There are many advanced application of SAS formats discussed in SAS documentation, books, and SUGI proceedings. Its application greatly simplified SAS coding in many cases. Most SAS programmers can create a SAS format with PROC FORMAT. However, managing existing SAS format library is more difficult.

EI simplified the process. As shown in Figure 21, one can select a SAS format library and list the names of the formats contained in the library. When a format is selected, its content is displayed as a SAS program

(PROC FORMAT). Then the user can turn on the Edit Mode and revise the SAS PROC FORMAT program. Click on the Save button will run the program and thus update the format library. In this way, EI provides a easy way to view and modify a SAS format. The displayed program can also be copied and used in any SAS programs.

Email: johnxu@1st-consulting.com

SAS ® is a registered trademark of SAS Institute, Inc. in the USA and Other countries.

Other brand and product names are registered trademarks or trademarks of their respective companies.

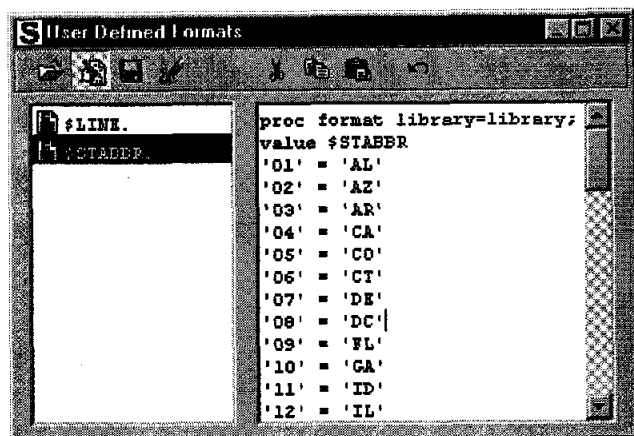


Figure 21 - User Defined SAS Format Library

CONCLUSION

Easy Interface for SAS is data exploring tool. Its user friendliness and other features enable us to focus on the business issue rather than on the programming issue. The technology used in the development can easily be adopted for implementing SAS applications.

REFERENCES

- ¹ Zhuan (John) Xu, (1998), Using SAS as an Automation Server in Windows Application Development, *Proceedings of the 23rd Annual SAS Users Group International Conference*, Nashville, Tennessee.
- ² SAS Institute Inc., *A formula for OLAP Success*,
- ³ Peter R. Welbrock, (1998), *Strategic Warehousing Principles Using SAS Software*, Cary, NC: SAS Institute Inc.
- ⁴ Zhuan (John) Xu, (1997), SAS Application Development Using Windows RAD Software for Front End, *Proceedings of the 8th Annual Conference of the MidWest SAS Users Group*, Chicago, Illinois.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Zhuan (John) Xu
4652 Valley View Ln
West Des Moines, IA 50225
Phone: (515) 778-4093