

How American Express Saved \$1M in CPU charges

Hermes Villalobos, American Express, Co., Miami Lakes, FL

ABSTRACT

There are many advantages of using code generators, for example, reducing the number of programming hours, thus allowing the SAS users to focus on analysis. Another example, in cases like this, is reducing the cost of CPU.

This project was designed with interaction on multiple platforms: Unix, Windows, and Mainframe. It is difficult to synchronize the programs running on these different platforms. In order to achieve some harmony among the programs, CRT, Microsoft Access, and FTP scripts were incorporated in to the project.

This paper will describe how to create a code generator and how to solve the problems related to the multiple platforms and multiple users.

INTRODUCTION

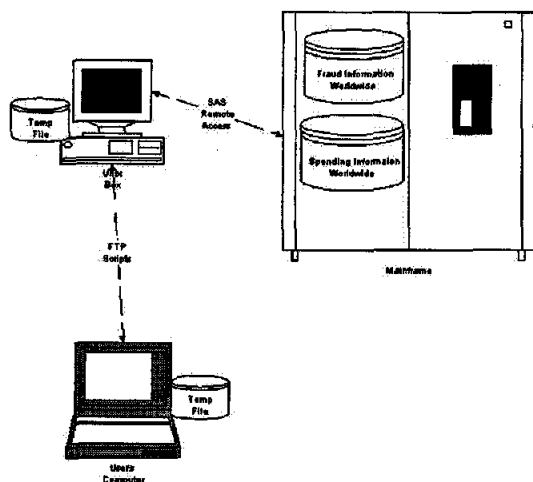
In 1997, the author was working as Senior Software Engineer for Fraud Prevention, analyzing large volumes of data from multiple sources and from multiple markets. All these analyses requests had something in common: all the layouts were pre-defined. Before starting a detailed analysis the author had to do some summarization of his information usually by country, city or industry. Then the author had to create a dataset with the information that he needed, then several procedures had to be run in order to analyze his information.

Next, the author researched and automated, as much as he could, the information extraction and analysis processes.

The author created a graphical user interface and SAS code generator in Microsoft Access. To transmit the programs to the Unix box the author used FTP scripts, and to submit the programs from Unix to the mainframe the author used CRT scripts.

ARCHITECTURE OF THE TOOL

The graphical user interface was located on the user's PC, the SAS programs ran on the Unix Platform, and the data repository was located on the mainframe as flat files.



HOW THE TOOL WORKS

The user interface generates a SAS program according to the user's input.

Then the user interface generates FTP script to transfer the program from the PC to the Unix Box.

Once the program is in the Unix box the user interface generates CRT script to submit the SAS program from the Unix.

This SAS program generates a temporary file with all the information requested. This is information that would be summarized or that could be the actual records from the databases.

Once the information arrives to the user's PC the information can be manipulated or exported to other media such as Excel, a SAS dataset in Unix, a flat file on the mainframe, or a comma-delimited file on the user's PC for further research.

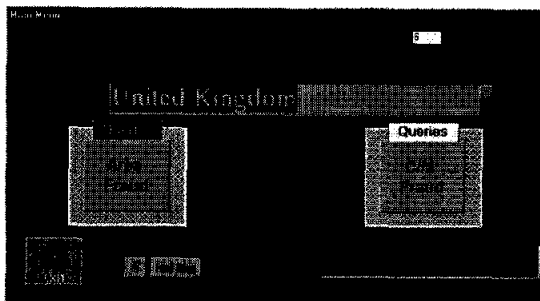
SAS REMOTE ACCESS: ADVANTAGES AND DISADVANTAGES

This tool generates and executes SAS programs in Unix. These programs access flat files remotely located on the mainframe. The main advantage of using this technique is that the CPU charges are zero. No mainframe programs are executed.

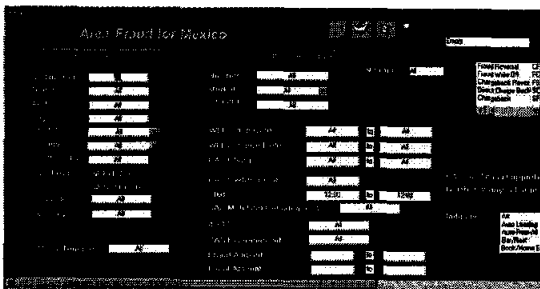
On the other hand, there is a restriction about the record length that can be accessed remotely. The author has developed tools with a maximum record length of 255 bytes and that still has good performance.

THE USER INTERFACE

The user interface allows the user to select the variables and criteria that the user needs to generate SAS programs according to need.



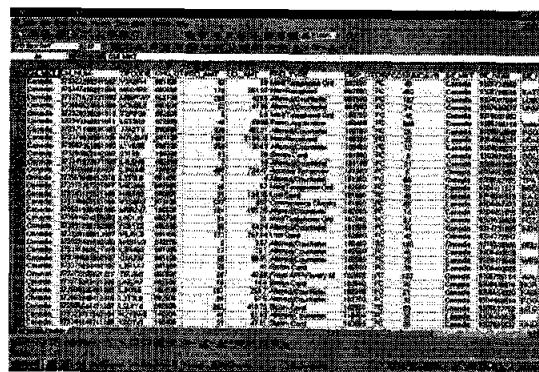
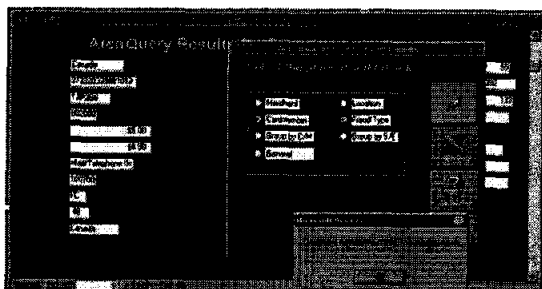
It also allows the user to focus on specific industries and cities.



The user interface was developed in Microsoft Access. The core component is the code generator function.

Code is generated in SAS, FTP, and CRT to submit the programs.

Once the information is received, the user can massage the information, analyze it and export it to Excel.



Above is a sample of information exported to Excel.

SAS CODE

The SAS code generated by the user interface uses the remote access method.

```
Filename my129 ftp "'FILENAME'"
host='IP ADDRESS' user='XXXXXX'
pass='XXXXXX'
RCMD='SITE RDW';
```

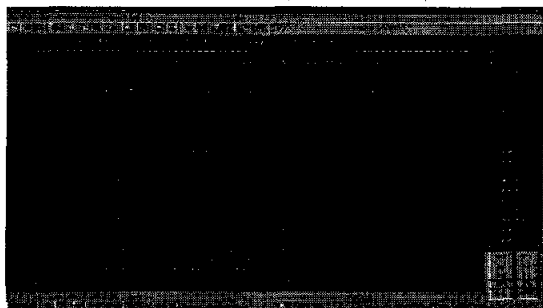
```
DATA nadal29 ;
  infile      my129      RECFM=F
  LRECL=298 ;
  INPUT
    @1  CM_MKT      $EBCDIC14.
    @15 CM_NUM      $EBCDIC15.
    @30 ZIPCODE     $EBCDIC10.
  ...
  ;
```

When a summarization is requested the SAS programs summarize the information using PROC SQL, and generate a flat file with a different layout.

DATA SOURCES ON MAINFRAME

All the information that the tool accesses, is based on flat files on the mainframe.

Here are some samples of such files. All the files have to be accessed since each one contains worldwide information.



IMPLICATIONS OF A MULTI USER TOOL

Since this tool was delivered worldwide, an algorithm was developed to avoid SAS programs and temp files from conflicting with one another.

The tool looks for the user's name on the system.ini file and generates all the programs and outputs based on the name found.

Since VILLA would be detected, for example, as the author's user name, then all the SAS programs and files are named using this name.

Here is a sample of FTP code generated to transmit a SAS program to Unix.

```
open IPAddress
User
Password
ascii
del VILLA.out
del VILLA.cas
```

```
del VILLA.in
del VILLA.tst
put c:\fast\fast.tst VILLA.tst
del VILLA.tst.log
bye
```

HOW THE TOOL SUBMITS A UNIX PROGRAM

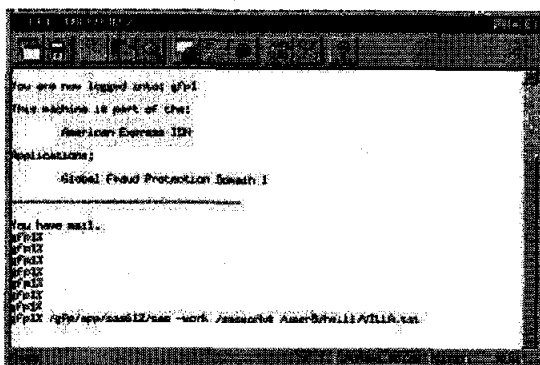
As mentioned before the tool generates CRT code to submit the Unix program.

CRT it is a terminal emulator that allows the user to run scripts. It was developed by Van Dyke technologies.

Here is sample of CRT code.

```
// login1.csf
expect( "ogin: ");
send("MyUser\r");
expect( "assword: " );
send( "MyPass\r" );
expect( "gfp1% " );
send( "\r" );
send( "/gfp/app/sas612/sas -work
/saswork4
/user5/hvill/VILLA.tst\r" );
expect( "gfp1% " );
send( "\r" );
send("logout\r");
Disconnect();
CloseWindow();
```

This code logs in to the Unix box and submits the SAS program generated by the tool, in this case, VILLA.tst.



CONCLUSION

The code generators can significantly reduce the programming time meaning increase in productivity when analyzing data.

The remote FTP data access provided by SAS reduces to zero the CPU charges. This is particularly important in companies that must pay for every time they submit a JOB. The author has been using this method for 4 years and has never received a CPU charge.

Finally, you have to be careful with the input data formats when transferring data from one platform to another.

REFERENCES

Hermes Villalobos, May 2001